aws

Guida per l'utente

Amazon Simple Storage Service

Versione API 2006-03-01

Amazon Simple Storage Service: Guida per l'utente

Table of Contents

Che cos'è Amazon S3?

Amazon Simple Storage Service (Amazon S3) è un servizio di archiviazione di oggetti che offre scalabilità, disponibilità dei dati, sicurezza e prestazioni tra le migliori del settore. I clienti di tutte le dimensioni e settori possono utilizzare Amazon S3 per archiviare e proteggere qualsiasi quantità di dati in un'ampia gamma di casi d'uso, come data lake, siti Web, applicazioni mobili, backup e ripristino, archivi, applicazioni per aziende, dispositivi IoT e analisi dei Big Data. Amazon S3 offre caratteristiche di gestione che consentono di ottimizzare, organizzare e configurare l'accesso ai dati per soddisfare specifici requisiti aziendali, organizzativi e di conformità.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Argomenti

Caratteristiche di Amazon S3

Classi di archiviazione

Amazon S3 offre una gamma di classi di storage concepite per i diversi casi d'uso, Ad esempio, è possibile archiviare dati di produzione essenziali su S3 Standard o S3 Express One Zone per accedervi più spesso, risparmiare sui costi archiviando i dati a cui si accede raramente in S3 Standard-IA o S3 One Zone-IA e archiviare i dati al minor costo in S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive.

Amazon S3 Express One Zone è una classe di archiviazione Amazon S3 a zona singola ad alte prestazioni, creata appositamente per fornire un accesso ai dati coerente di pochi millisecondi per le applicazioni sensibili alla latenza. S3 Express One Zone è la classe di cloud object storage con la latenza più bassa disponibile oggi, con velocità di accesso ai dati fino a 10 volte più elevate e con costi di richiesta inferiori del 50% rispetto a S3 Standard. S3 Express One Zone è la prima classe di archiviazione S3 in cui è possibile selezionare una singola zona di disponibilità con la possibilità di co-ubicare l'archiviazione di oggetti con le risorse di calcolo, che offre la massima velocità di accesso possibile. Inoltre, per aumentare ulteriormente la velocità di accesso e supportare centinaia di migliaia di richieste al secondo, i dati vengono archiviati in un nuovo tipo di bucket: un bucket di directory Amazon S3. Per ulteriori informazioni, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Puoi archiviare i dati con modelli di accesso mutevoli o sconosciuti in S3 Intelligent-Tiering, una classe che ottimizza i costi di archiviazione spostando automaticamente i dati tra quattro livelli di accesso quando cambiano i relativi modelli. Questi quattro livelli di accesso includono due livelli di accesso a bassa latenza ottimizzati per l'accesso frequente e sporadico e due livelli di accesso all'archivio progettati per l'accesso asincrono e per dati a cui accedi raramente.

Per ulteriori informazioni, consulta Utilizzo delle classi di storage di Amazon S3.

Gestione dello storage

Amazon S3 dispone di caratteristiche di gestione dell'archiviazione utilizzabili per gestire i costi, rispettare i requisiti normativi, ridurre la latenza e salvare più copie distinte dei dati per soddisfare i requisiti di conformità.

- Ciclo di vita S3: consente di impostare la configurazione del ciclo di vita per gestire gli oggetti e archiviarli all'insegna dell'efficienza in termini di costi durante l'intero ciclo di vita. Puoi spostare gli oggetti in altre classi di archiviazione S3 o far scadere oggetti che raggiungono la fine del loro ciclo.
- Blocco degli oggetti S3: impedisci che un oggetto di Amazon S3 venga eliminato o sovrascritto per un determinato periodo di tempo o in modo indefinito. Puoi utilizzare Object Lock per soddisfare i requisiti normativi che richiedono lo storage write-once-read-many(WORM) o semplicemente per aggiungere un altro livello di protezione contro le modifiche e le eliminazioni degli oggetti.
- Replica S3: replica gli oggetti e i rispettivi metadati e tag degli oggetti in uno o più bucket di destinazione uguali o diversi Regioni AWS per ridurre latenza, conformità, sicurezza e altri casi d'uso.
- Operazioni in batch S3: consente di gestire qualsiasi numero di oggetti su larga scala con una singola richiesta API S3 o pochi clic nella console di Amazon S3. È possibile utilizzare Batch

Operations per eseguire operazioni come Copy, Invoke AWS Lambda e Restore su milioni o miliardi di oggetti.

Gestione degli accessi e sicurezza

Amazon S3 offre caratteristiche per la verifica e la gestione degli accessi ai tuoi bucket e oggetti. Per impostazione predefinita, i bucket S3 e gli oggetti al loro interno sono privati. Puoi accedere solo alle risorse S3 che hai creato. Per concedere autorizzazioni granulari delle risorse che supportano il tuo caso d'uso specifico o per verificare le autorizzazioni delle tue risorse Amazon S3, puoi utilizzare le seguenti caratteristiche.

- Blocco dell'accesso pubblico di S3: blocca l'accesso pubblico a bucket S3 e oggetti. Per impostazione predefinita, le impostazioni Blocco dell'accesso pubblico sono attivate a livello di bucket. È consigliabile mantenere tutte le impostazioni Blocco dell'accesso pubblico disabilitate, a meno che non sia necessario disattivarne una o più di una per il caso d'uso specifico. Per ulteriori informazioni, consulta Configurazione delle impostazioni di blocco dell'accesso pubblico per i bucket S3.
- AWS Identity and Access Management (IAM): IAM è un servizio Web che consente di controllare in modo sicuro l'accesso alle AWS risorse, incluse le risorse Amazon S3. Con IAM, puoi gestire centralmente le autorizzazioni che controllano le AWS risorse a cui gli utenti possono accedere. Utilizza IAM per controllare chi è autenticato (accesso effettuato) e autorizzato (dispone di autorizzazioni) per l'utilizzo di risorse.
- Policy di bucket: utilizza il linguaggio delle policy basato su IAM per configurare le autorizzazioni basate sulle risorse per i bucket S3 e gli oggetti in essi contenuti.
- Punto di accesso Amazon S3: configura gli endpoint di rete denominati con policy di accesso dedicate per gestire l'accesso ai dati su vasta scala per set di dati condivisi in Amazon S3.
- Liste di controllo degli accessi (ACL): concedi autorizzazioni di lettura e scrittura per singoli bucket e oggetti agli utenti autorizzati. Come regola generale, è consigliabile utilizzare policy basate sulle risorse S3 (policy di bucket e policy dei punti di accesso) o policy utente IAM per il controllo degli accessi anziché ACL. Le policy sono un'opzione di controllo degli accessi semplificata e più flessibile. Con le policy dei bucket e le policy dei punti di accesso, puoi definire le regole applicabili globalmente a tutte le richieste alle risorse Amazon S3. Per ulteriori informazioni su casi specifici quando desideri utilizzare le ACL anziché le policy basate sulle risorse o le policy utente IAM, consultare Gestione degli accessi con le ACL.
- S3 Proprietà dell'oggetto: consente di assumere la proprietà di ogni oggetto nel bucket, semplificando la gestione degli accessi per i dati archiviati in Amazon S3. S3 Proprietà dell'oggetto

è un'impostazione a livello di bucket di Amazon S3 che puoi utilizzare per disabilitare o abilitare le ACL. Per impostazione predefinita, le ACL sono disabilitate. Con le ACL disabilitate, il proprietario del bucket possiede tutti gli oggetti nel bucket e gestisce l'accesso ai dati in maniera esclusiva utilizzando policy di gestione dell'accesso.

- IAM Access Analyzer per S3: valuta e monitora le policy di accesso al bucket S3, assicurando che forniscano solo l'accesso previsto alle risorse S3.

Elaborazione dei dati

Per trasformare i dati e attivare i flussi di lavoro in modo che automatizzino una serie di altre attività di elaborazione su larga scala, puoi utilizzare le seguenti caratteristiche.

- Lambda dell'oggetto S3: aggiungi il tuo codice alle richieste GET, HEAD e LIST di S3 per modificare ed elaborare i dati quando vengono restituiti a un'applicazione. Questa caratteristica consente di filtrare righe, ridimensionare dinamicamente immagini, oscurare dati riservati e molto altro ancora.
- Notifiche di eventi: attiva flussi di lavoro che utilizzano Amazon Simple Notification Service (Amazon SNS), Amazon Simple Queue Service (Amazon SQS) e quando viene apportata una modifica alle tue risorse S3. AWS Lambda

Registrazione e monitoraggio dell'archiviazione

Amazon S3 fornisce strumenti di registrazione e monitoraggio che puoi utilizzare per monitorare e controllare come vengono utilizzate le tue risorse Amazon S3. Per ulteriori informazioni, Strumenti di monitoraggio.

Strumenti di monitoraggio automatici

- CloudWatchParametri Amazon per Amazon S3: monitora lo stato operativo delle tue risorse S3 e configura avvisi di fatturazione quando gli addebiti stimati raggiungono una soglia definita dall'utente.
- AWS CloudTrail— Registra le azioni intraprese da un utente, da un ruolo o da un utente Servizio AWS in Amazon S3. CloudTrail i log forniscono un tracciamento dettagliato delle API per le operazioni S3 a livello di bucket e a livello di oggetto.

Strumenti di monitoraggio manuali

- Registrazione degli accessi al server: fornisce registri dettagliati per le richieste effettuate a un bucket. Puoi utilizzare i registri di accesso al server per molti casi d'uso, come eseguire verifiche di sicurezza e accesso, conoscere la tua base clienti o capire meglio la fattura Amazon S3.
- AWS Trusted Advisor: valuta il tuo account utilizzando i controlli delle AWS migliori pratiche per identificare modi per ottimizzare l' AWS infrastruttura, migliorare la sicurezza e le prestazioni, ridurre i costi e monitorare le quote di servizio. Puoi quindi seguire i suggerimenti per ottimizzare i servizi e le risorse.

Analisi dei dati e informazioni dettagliate

Amazon S3 offre caratteristiche per aiutarti a ottenere visibilità sull'utilizzo dello spazio di archiviazione, che ti consente di comprendere, analizzare e ottimizzare meglio lo spazio di archiviazione su larga scala.

- Amazon S3 Storage Lens: consente di comprendere, analizzare e ottimizzare l'archiviazione. S3 Storage Lens offre oltre 60 metriche di utilizzo e attività e dashboard interattivi per aggregare i dati per l'intera organizzazione, account specifici, bucket o prefissi. Regioni AWS
- Analisi della classe di storage: consente di analizzare i modelli di accesso all'archiviazione per decidere quando è il momento di spostare i dati in una classe più conveniente.
- S3 Inventory con report di Inventory: consente di verificare e creare report sugli oggetti e sui relativi metadati e configurare altre caratteristiche di Amazon S3 per intervenire sui report di Inventory. Ad esempio, puoi creare report sullo stato di replica e crittografia degli oggetti. Per un elenco di tutti i metadata disponibili per ogni oggetto nei report di Amazon S3 Inventory, consulta questa sezione.

Forte coerenza

In generale, Amazon S3 offre una forte read-after-write coerenza per le richieste PUT e DELETE degli oggetti nel bucket Amazon S3. Regioni AWS Questo comportamento vale sia per le scritture dei nuovi oggetti che per le richieste PUT che sovrascrivono gli oggetti esistenti e le richieste DELETE. Inoltre, le operazioni di lettura su Amazon S3 Select, le liste di controllo accessi (ACL) Amazon S3, i tag oggetto Amazon S3 e i metadati degli oggetti (ad esempio, l'oggetto HEAD) sono fortemente coerenti. Per ulteriori informazioni, consulta Modello di consistenza dati Amazon S3.

Come funziona Amazon S3

Amazon S3 è un servizio che consente di archiviare dati come oggetti nei bucket. Un oggetto è un file e tutti i metadati che lo descrivono. Un bucket è un container per oggetti o file.

Per archiviare dati in Amazon S3, per prima cosa devi creare un bucket e specificarne nome e Regione AWS, quindi devi caricare i dati nel bucket come oggetti in Amazon S3. Ogni oggetto contiene una chiave (o nome chiave), che è l'identificatore univoco dell'oggetto nel bucket.

S3 fornisce funzionalità che puoi configurare per supportare il tuo caso d'uso specifico. Puoi utilizzare Controllo delle versioni S3 per mantenere più versioni di un oggetto in un unico bucket e consentire di ripristinare oggetti che vengono accidentalmente eliminati o sovrascritti.

I bucket e gli oggetti che contengono sono privati e accessibili solo se concedi esplicitamente le autorizzazioni di accesso. Puoi utilizzare bucket policy, policy AWS Identity and Access Management (IAM), liste di controllo degli accessi (ACL) e punti di accesso S3 per gestire l'accesso.

Argomenti

Bucket

Un bucket è un container per gli oggetti archiviati in Amazon S3. Puoi archiviare un numero qualsiasi di oggetti in un bucket e avere fino a 100 bucket nel tuo account. Per richiedere un aumento, visita la Console Service Quotas.

Ogni oggetto è contenuto in un bucket. Ad esempio, se l'oggetto denominato `photos/puppy.jpg` è archiviato nel bucket `DOC-EXAMPLE-BUCKET` nella regione Stati Uniti occidentali (Oregon), è

indirizzabile tramite l'URL `https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/photos/puppy.jpg`. Per ulteriori informazioni, consulta Accesso a un bucket.

Quando crei un bucket, inserisci un nome e scegli la Regione AWS dove si troverà. Dopo avere creato un bucket, non puoi modificarne il nome né la regione. I nomi dei bucket devono seguire queste regole di denominazione. Puoi inoltre configurare un bucket per l'uso del controllo versioni di S3 o altre caratteristiche per la gestione dell'archiviazione

I bucket inoltre:

- Organizzano lo spazio dei nomi Amazon S3 al livello più alto.
- Identificano l'account responsabile del costo di archiviazione e trasferimento dati.
- Forniscono opzioni di controllo degli accessi, come policy di bucket, ACL e punti di accesso S3, utilizzabili per gestire l'accesso alle risorse di Amazon S3.
- Servono come unità di aggregazione per i report di utilizzo.

Per ulteriori informazioni sui bucket, consulta Panoramica dei bucket.

Oggetti

Gli oggetti sono le entità fondamentali archiviate in Amazon S3 e sono composti da dati e metadata. I metadati sono invece un set di coppie nome-valore che descrivono l'oggetto. Queste coppie includono alcuni metadati di default, ad esempio la data dell'ultima modifica, e metadati HTTP standard, come `Content-Type`. È anche possibile specificare metadata personalizzati al momento dell'archiviazione dell'oggetto.

Un oggetto viene identificato in modo univoco in un bucket tramite un (nome) chiave e un ID versione (se Controllo delle versioni S3 è abilitato nel bucket). Per ulteriori informazioni sugli oggetti, consulta Panoramica degli oggetti di Amazon S3.

Chiavi

Una chiave oggetto (o nome chiave) è l'identificatore univoco di un oggetto in un bucket. Per ogni oggetto in un bucket è presente esattamente una chiave. La combinazione di bucket, chiave oggetto e, facoltativamente, ID versione (se il Controllo delle versioni S3 è abilitato per il bucket) identificherà in modo univoco ogni oggetto. Quindi puoi pensare ad Amazon S3 come a una mappa di dati di base tra "bucket + chiave + versione" e l'oggetto stesso.

Si può fare riferimento in modo univoco a ogni oggetto in Amazon S3 tramite la combinazione di endpoint del servizio Web, nome del bucket, chiave e, facoltativamente, una versione. Ad esempio, nell'URL `https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/photos/puppy.jpg`, `DOC-EXAMPLE-BUCKET` è il nome del bucket e `photos/puppy.jpg` è la chiave.

Per ulteriori informazioni sulle chiavi degli oggetti, consulta Creazione di nomi di chiavi oggetto.

Funzione Controllo delle versioni S3

Puoi utilizzare Controllo delle versioni S3 per conservare più versioni di un oggetto nello stesso bucket. Puoi utilizzare Controllo delle versioni S3 per conservare, recuperare e ripristinare qualsiasi versione di ogni oggetto archiviato nei bucket . Puoi facilmente eseguire il ripristino dopo errori dell'applicazione e operazioni non intenzionali dell'utente.

Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

ID versione

Se abiliti Controllo delle versioni S3 per un bucket, Amazon S3 genera un ID versione univoco per tutti gli oggetti aggiunti a tale bucket. Gli oggetti già esistenti nel bucket al momento dell'attivazione del controllo delle versioni hanno un ID versione `null`. Se modifichi questi (o altri) oggetti con altre operazioni, come CopyObjecte PutObject, i nuovi oggetti ottengono un ID di versione univoco.

Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Policy del bucket

Una bucket policy è una policy basata sulle risorse AWS Identity and Access Management (IAM) che puoi utilizzare per concedere le autorizzazioni di accesso al tuo bucket e agli oggetti in esso contenuti. Solo il proprietario del bucket può associare una policy a un bucket. Le autorizzazioni allegate a un bucket si applicano a tutti gli oggetti del bucket di proprietà del proprietario del bucket. Le policy di bucket sono limitate a una dimensione di 20 KB.

Le policy di bucket utilizzano la sintassi delle policy di accesso basata su JSON, che è lo standard di AWS. Puoi utilizzare policy di bucket per aggiungere o negare autorizzazioni per gli oggetti in un bucket. Le policy di bucket approvano o negano le richieste in base agli elementi che contengono, inclusi richiedente, operazioni S3, risorse e aspetti o condizioni della richiesta (ad esempio, l'indirizzo IP utilizzato per inviarla). Ad esempio, puoi creare una policy che conceda autorizzazioni tra account per caricare oggetti in un bucket S3 garantendo al contempo che il proprietario del bucket abbia il

pieno controllo degli oggetti caricati. Per ulteriori informazioni, consulta Esempi di policy relative ai bucket di Amazon S3.

Nella policy di bucket puoi utilizzare caratteri jolly negli Amazon Resource Name (ARN) e altri valori per concedere autorizzazioni a un sottoinsieme di oggetti. Ad esempio, puoi controllare l'accesso a gruppi di oggetti che iniziano con un prefisso comune o terminano con una determinata estensione, come `.html`.

Punto di accesso S3

I punti di accesso Amazon S3 sono endpoint di rete denominati con policy di accesso dedicate che descrivono come è possibile accedere ai dati utilizzando tale endpoint. Gli access point sono collegati a bucket che puoi utilizzare per eseguire operazioni sugli oggetti S3, come e. GetObject PutObject I punti di accesso semplificano la gestione dell'accesso ai dati su vasta scala per set di dati condivisi in Amazon S3.

Ogni punto di accesso ha una propria policy. Puoi inoltre configurare impostazioni di blocco dell'accesso pubblico personalizzate per ciascun punto di accesso. Per limitare l'accesso ai dati di Amazon S3 a una rete privata puoi configurare qualsiasi punto di accesso per accettare le richieste solo da un virtual private cloud (VPC).

Per ulteriori informazioni, consulta Gestione dell'accesso ai dati con Punti di accesso Amazon S3.

Liste di controllo degli accessi (ACL)

Puoi utilizzare le ACL per concedere autorizzazioni di lettura e scrittura per bucket e oggetti singoli agli utenti autorizzati. A ogni bucket e oggetto è allegata una ACL come sottorisorsa. L'ACL definisce a quali Account AWS o gruppi è concesso l'accesso e il tipo di accesso. Le ACL sono un meccanismo di controllo degli accessi che precede IAM. Per ulteriori informazioni sulle ACL, consulta Panoramica delle liste di controllo accessi (ACL).

S3 Proprietà dell'oggetto è un'impostazione a livello di bucket Amazon S3 che è possibile utilizzare per controllare la proprietà degli oggetti caricati nel bucket e per disabilitare o abilitare le liste di controllo degli accessi (ACL). Per impostazione predefinita, Proprietà dell'oggetto è impostata su Proprietario del bucket applicato e tutte le liste di controllo degli accessi (ACL) sono disabilitate. Quando le ACL sono disabilitate, il proprietario del bucket dispone di tutti gli oggetti nel bucket e gestisce l'accesso ad essi in maniera esclusiva utilizzando policy di gestione dell'accesso.

La maggior parte degli attuali casi d'uso in Amazon S3 non richiede più l'uso delle ACL. È consigliabile mantenere le ACL disabilitate, tranne nelle circostanze in cui è necessario controllare

individualmente l'accesso per ciascun oggetto. Con le ACL disabilitate, puoi utilizzare le policy per controllare l'accesso a tutti gli oggetti nel bucket, a prescindere da chi ha caricato gli oggetti nel bucket. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

Regioni

Puoi scegliere l'area geografica Regione AWS in cui Amazon S3 archivia i bucket che crei. La scelta di una regione permette di ottimizzare la latenza, ridurre al minimo i costi o rispondere ai requisiti normativi. Gli oggetti archiviati in un'altra regione Regione AWS non escono mai dalla regione a meno che non vengano trasferiti o replicati esplicitamente in un'altra regione. Ad esempio, gli oggetti archiviati nella regione Europa (Irlanda) non lasceranno mai tale regione.

> **Note**
>
> Puoi accedere ad Amazon S3 e alle sue funzionalità solo nelle versioni abilitate per il Regioni AWS tuo account. Per ulteriori informazioni sull'abilitazione di una regione per la creazione e la gestione di AWS risorse, consulta Managing Regioni AWS in the Riferimenti generali di AWS.

Per un elenco degli endpoint e delle regioni Amazon S3 disponibili, consultare la sezione relativa a regioni ed endpoint nella Riferimenti generali di AWS.

Modello di consistenza dati Amazon S3

In generale, Amazon S3 offre una forte read-after-write coerenza per le richieste PUT e DELETE degli oggetti nel bucket Amazon S3. Regioni AWS Questo comportamento vale sia per le scritture di nuovi oggetti che per le richieste PUT che sovrascrivono gli oggetti esistenti e le richieste DELETE. Inoltre, le operazioni di lettura su Amazon S3 Select, le liste di controllo accessi Amazon S3, i tag oggetto Amazon S3 e i metadati degli oggetti (ad esempio, l'oggetto HEAD) sono fortemente coerenti.

Gli aggiornamenti a una singola chiave sono atomici. Ad esempio, se esegui una richiesta PUT su una chiave esistente da un thread ed esegui poi una richiesta GET sulla stessa chiave da un secondo thread contemporaneamente, otterrai i vecchi dati o i nuovi dati, ma mai dati parziali o danneggiati.

Amazon S3 ottiene un'alta disponibilità replicando i dati su più server in data center AWS . Se una richiesta PUT ha esito positivo, i dati verranno archiviati in totale sicurezza. Qualsiasi lettura

(richiesta GET o LIST) avviata dopo la ricezione di una risposta PUT riuscita restituirà i dati scritti dall'operazione PUT. Di seguito sono riportati alcuni esempi di questo comportamento.

- Un processo scrive un nuovo oggetto in Amazon S3 ed elenca immediatamente le chiavi nel relativo bucket. Il nuovo oggetto viene visualizzato nell'elenco.
- Un processo sostituisce un oggetto esistente e tenta immediatamente di effettuarne la lettura. Amazon S3 restituisce i nuovi dati.
- Un processo elimina un oggetto esistente e tenta immediatamente di effettuarne la lettura. Amazon S3 non restituisce alcun dato poiché l'oggetto è stato eliminato.
- Un processo elimina un oggetto esistente ed elenca immediatamente le chiavi nel relativo bucket. L'oggetto non viene visualizzato nell'elenco.

> **Note**
>
> - Amazon S3 non supporta il blocco degli oggetti per istanze di scrittura simultanee. Se vengono effettuate simultaneamente due richieste PUT per la stessa chiave, la richiesta con l'ultimo timestamp ha la precedenza. Se questo rappresenta un problema, devi creare un meccanismo di blocco degli oggetti nell'applicazione.
> - Gli aggiornamenti sono basati su chiave. Non è possibile eseguire aggiornamenti atomici tra le chiavi. Non si può ad esempio eseguire l'aggiornamento di una chiave dipendente dall'aggiornamento di un'altra chiave, a meno che non si progetti questa funzionalità nell'applicazione.

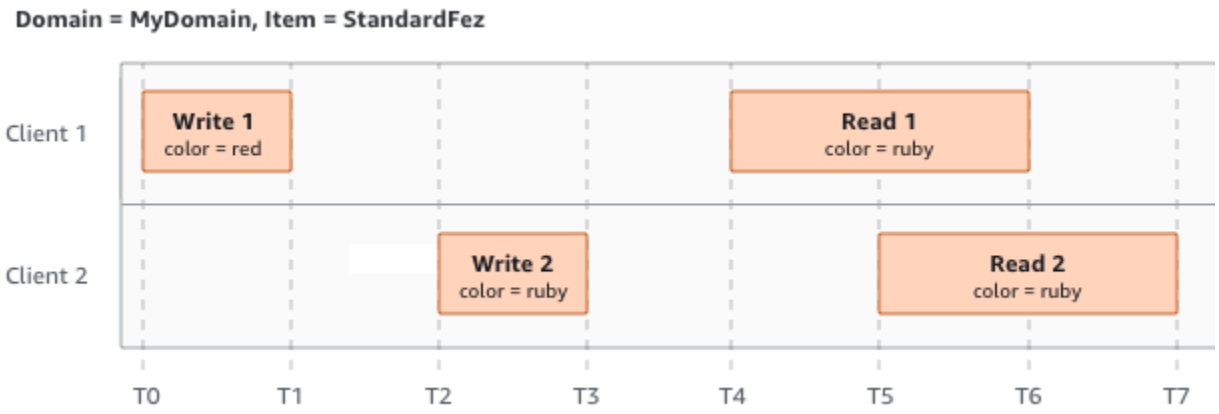
Le configurazioni dei bucket hanno un modello di consistenza. In particolare, questo significa che:

- Se elimini un bucket e visualizzi immediatamente tutti i bucket, il bucket eliminato potrebbe comunque essere visualizzato nell'elenco.
- Se abiliti il controllo delle versioni su un bucket per la prima volta, potrebbe essere necessario un breve periodo di tempo per la propagazione completa della modifica. Suggeriamo di attendere 15 minuti dopo aver abilitato il controllo delle versioni prima di eseguire operazioni di scrittura (richieste PUT o DELETE) sugli oggetti nel bucket.

Applicazioni simultanee

In questa sezione sono riportati esempi di comportamento previsto da Amazon S3 quando più client scrivono sugli stessi articoli.

In questo esempio, entrambe le richieste di scrittura W1 e W2 terminano prima dell'avvio delle letture R1 e R2. Poiché S3 è fortemente consistente, R1 e R2 restituiscono entrambi `color = ruby`.



Nell'esempio successivo, la scrittura W2 non termina prima dell'avvio della lettura R1. Pertanto, R1 potrebbe restituire `color = ruby` o `color = garnet`. Tuttavia, dal momento che W1 e W2 terminano prima dell'inizio di R2, R2 restituisce `color = garnet`.

Nell'ultimo esempio, W2 inizia prima che W1 abbia ricevuto una notifica. Pertanto, queste scritture sono considerate simultanee. Amazon S3 utilizza internamente la last-writer-wins semantica per determinare quale scrittura ha la precedenza. Tuttavia, l'ordine in cui Amazon S3 riceve le richieste e l'ordine in cui le applicazioni ricevono le notifiche non possono essere previsti a causa di vari fattori quali la latenza della rete. Ad esempio, W2 potrebbe essere avviata da un'istanza Amazon EC2 nella stessa regione, mentre W1 potrebbe essere avviata da un host più lontano. Il modo migliore

per determinare il valore finale è eseguire una lettura dopo che entrambe le scritture sono state riconosciute.

Servizi correlati

Dopo aver caricato i dati in Amazon S3, puoi utilizzarli con altri AWS servizi. Di seguito vengono riportati i servizi che potresti utilizzare più di frequente:

- Amazon Elastic Compute Cloud (Amazon EC2): fornisce capacità di calcolo scalabile e sicura in Cloud AWS. L'utilizzo di Amazon EC2 elimina la necessità di investimenti anticipati in hardware e ti permette di sviluppare e implementare più rapidamente le applicazioni. Puoi utilizzare Amazon EC2 per avviare il numero di server virtuali necessari, configurare la sicurezza e i servizi di rete, nonché gestire l'archiviazione.
- Amazon EMR: consente ad aziende, ricercatori, analisti e sviluppatori di elaborare un'enorme quantità di dati in modo semplice ed economico. Amazon EMR usa un framework Hadoop gestito eseguito sull'infrastruttura a livello Web di Amazon EC2 e Amazon S3.
- AWS Snow Family: aiuta i clienti che devono eseguire operazioni in ambienti austeri, non basati su data center e in luoghi in cui manca una connettività di rete coerente. È possibile utilizzare i dispositivi AWS Snow Family per accedere localmente e in modo conveniente alla potenza di archiviazione e di calcolo di Internet Cloud AWS in luoghi in cui una connessione Internet potrebbe non essere un'opzione.
- AWS Transfer Family: fornisce supporto completamente gestito per i trasferimenti di file diretti in entrata e uscita da Amazon S3 o Amazon Elastic File System (Amazon EFS) utilizzando i protocolli SFTP, FTPS e FTP.

Accesso ad Amazon S3

Puoi lavorare con Amazon S3 nei modi descritti di seguito:

AWS Management Console

La console è un'interfaccia utente basata sul Web per la gestione di Amazon S3 AWS e delle risorse. Se ti sei registrato a Account AWS, puoi accedere alla console Amazon S3 accedendo AWS Management Console e scegliendo S3 dalla AWS Management Console home page.

AWS Command Line Interface

Puoi usare gli strumenti della AWS riga di comando per impartire comandi o creare script dalla riga di comando del tuo sistema per eseguire attività AWS (incluso S3).

Il AWS Command Line Interface (AWS CLI) fornisce comandi per un ampio set di. Servizi AWS AWS CLI È supportato su Windows, macOS e Linux. Per iniziare, consulta la Guida per l'utente di AWS Command Line Interface . Per ulteriori informazioni sui comandi per Amazon S3, consulta s3api e s3control nella pagina di riferimento dei comandi della AWS CLI .

AWS SDK

AWS fornisce SDK (kit di sviluppo software) costituiti da librerie e codice di esempio per vari linguaggi e piattaforme di programmazione (Java, Python, Ruby, .NET, iOS, Android e così via). Gli AWS SDK offrono un modo conveniente per creare l'accesso programmatico a S3 e. AWS Amazon S3 è un servizio REST. Puoi inviare richieste ad Amazon S3 utilizzando le librerie AWS SDK, che racchiudono l'API REST di Amazon S3 sottostante e semplificano le attività di programmazione. Ad esempio, gli SDK si occupano di attività quali il calcolo delle firme, la firma crittografica delle richieste, la gestione degli errori e la ripetizione automatica delle richieste. Per informazioni sugli AWS SDK, incluso come scaricarli e installarli, consulta Tools for. AWS

Ogni interazione con Amazon S3 è autenticata o anonima. Se utilizzi gli AWS SDK, le librerie calcolano la firma per l'autenticazione dalle chiavi che fornisci. Per ulteriori informazioni su come effettuare richieste ad Amazon S3, consulta Esecuzione di richieste.

API REST di Amazon S3

L'architettura di Amazon S3 è ideata per essere indipendente dal linguaggio di programmazione e per utilizzare le interfacce supportate da AWS per archiviare e recuperare oggetti. Puoi accedere a S3 e

AWS a livello di programmazione utilizzando l'API REST di Amazon S3. L'API REST è un'interfaccia HTTP per Amazon S3. Con l'API REST, uilizzi le richieste HTTP standard per creare, recuperare ed eliminare bucket e oggetti.

Per utilizzare l'API REST, puoi servirti di qualunque kit di strumenti in grado di supportare HTTP. Puoi anche utilizzare un browser per recuperare gli oggetti, purché siano leggibili in modo anonimo.

Poiché l'API REST utilizza codici di stato e intestazioni HTTP standard, i kit di strumenti e i browser standard funzionano come previsto. In alcune aree sono state aggiunte funzionalità ad HTTP, ad esempio le intestazioni per il supporto del controllo accessi. Le nuove funzionalità sono state in tali casi aggiunte in modo da essere conformi allo stile di utilizzo di HTTP standard.

Se effettui chiamate API REST direttamente dall'applicazione in uso, devi scrivere il codice per calcolare la firma e aggiungerlo alla richiesta. Per ulteriori informazioni su come effettuare richieste ad Amazon S3, consulta Esecuzione di richieste.

> **Note**
>
> Il supporto API SOAP su HTTP è obsoleto ma è ancora disponibile su HTTPS. Le funzioni più recenti di Amazon S3 non sono supportate per SOAP. Ti consigliamo di utilizzare l'API REST o gli AWS SDK.

Prezzi di Amazon S3

La determinazione dei prezzi di Amazon S3 è stata concepita in modo da non dover pianificare requisiti di storage per la tua applicazione. La maggior parte dei provider di archiviazione richiede l'acquisto di una quantità predeterminata di capacità di archiviazione e di trasferimento di rete. In questi casi, se superi questa capacità, il servizio viene disattivato o ti vengono addebitati costi aggiuntivi elevati. Se non si supera tale capacità, si pagherà comunque l'importo per l'intera capacità.

Con Amazon S3 si paga esclusivamente ciò che si utilizza, senza costi nascosti o aggiuntivi. Questo modello ti offre un servizio a costo variabile che può crescere con la tua azienda, offrendoti al contempo i vantaggi in termini di costi dell'infrastruttura. AWS Per ulteriori informazioni, consulta Prezzi di Amazon S3.

Quando ti registri AWS, il tuo Account AWS viene automaticamente registrato per tutti i servizi in AWS, incluso Amazon S3. Tuttavia, vengono addebitati solo i servizi che utilizzi. Se sei un nuovo

cliente Amazon S3, puoi iniziare a utilizzare Amazon S3 gratuitamente. Per ulteriori informazioni, consulta Piano gratuito AWS.

Per vedere la tua fattura, vai sul Pannello di controllo Gestione fatturazione e costi nella console AWS Billing and Cost Management. Per ulteriori informazioni sulla Account AWS fatturazione, consulta la Guida per l'AWS Billing utente. In caso di domande relative alla AWS fatturazione e Account AWS, contatta l'AWS assistenza.

Conformità PCI DSS

Amazon S3 supporta l'elaborazione, l'archiviazione e la trasmissione di dati di carte di credito da parte di un esercente o di un provider di servizi, oltre a essere conforme allo standard Payment Card Industry Data Security Standard (PCI DSS). Per ulteriori informazioni su PCI DSS, incluso come richiedere una copia del PCI AWS Compliance Package, vedere PCI DSS Level 1.

Nozioni di base su Amazon S3

Puoi iniziare con Amazon S3 lavorando con bucket e oggetti. Un bucket è un container per oggetti o file. Un oggetto è un file e tutti i metadati che descrivono tale file.

Per memorizzare un oggetto in Amazon S3, crei un bucket e quindi carichi l'oggetto in un bucket. Quando l'oggetto si trova nel bucket, è possibile aprirlo, scaricarlo e spostarlo. Quando non hai più bisogno di un oggetto o di un bucket, puoi ripulire le tue risorse.

Con Amazon S3 paghi solo per le risorse utilizzate. Per ulteriori informazioni sulle funzionalità e sui prezzi di Amazon S3, consulta Amazon S3. Se sei un nuovo cliente Amazon S3, puoi iniziare a utilizzare Amazon S3 gratuitamente. Per ulteriori informazioni, consulta Piano gratuito AWS.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Video: Nozioni di base su Amazon S3

Prerequisiti

Prima di iniziare, devi accertarti di avere completato le fasi in Prerequisito: Configurazione di Amazon S3.

Argomenti

Prerequisito: Configurazione di Amazon S3

Quando ti registri AWS, il tuo Account AWS viene automaticamente registrato per tutti i servizi in AWS, incluso Amazon S3. Ti vengono addebitati solo i servizi che utilizzi.

Con Amazon S3 paghi solo per le risorse utilizzate. Per ulteriori informazioni sulle funzionalità e sui prezzi di Amazon S3, consulta Amazon S3. Se sei un nuovo cliente Amazon S3, puoi iniziare a utilizzare Amazon S3 gratuitamente. Per ulteriori informazioni, consulta Piano gratuito AWS.

Per configurare Amazon S3, segui la procedura descritta nelle sezioni seguenti.

Quando ti registri AWS e configuri Amazon S3, puoi facoltativamente modificare la lingua di visualizzazione in. AWS Management Console Per ulteriori informazioni, consulta Modifica della lingua della AWS Management Console nella Guida introduttiva di AWS Management Console .

Argomenti

- Registrati per un Account AWS
- Crea un utente con accesso amministrativo

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la pagina all'indirizzo https://portal.aws.amazon.com/billing/signup.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come procedura consigliata in materia di sicurezza, assegnate l'accesso amministrativo a un utente e utilizzate solo l'utente root per eseguire attività che richiedono l'accesso da parte dell'utente root.

AWS ti invia un'e-mail di conferma dopo il completamento della procedura di registrazione. È possibile visualizzare l'attività corrente dell'account e gestire l'account in qualsiasi momento accedendo all'indirizzo https://aws.amazon.com/ e selezionando Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .
2. In IAM Identity Center, concedi l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con le impostazioni predefinite IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accedi come utente con accesso amministrativo

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso ad altri utenti

1. In IAM Identity Center, crea un set di autorizzazioni che segua la migliore pratica di applicazione delle autorizzazioni con privilegi minimi.

   Per istruzioni, consulta Creare un set di autorizzazioni nella Guida per l'utente.AWS IAM Identity Center
2. Assegna gli utenti a un gruppo, quindi assegna l'accesso Single Sign-On al gruppo.

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente.AWS IAM Identity Center

Fase 1: creare il primo bucket S3

Dopo la registrazione AWS, sei pronto per creare un bucket in Amazon S3 utilizzando il. AWS Management Console Ogni oggetto in Amazon S3 viene archiviato in un bucket. Prima di poter archiviare dati in Amazon S3, devi creare un bucket.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

> **Note**
>
> Non è previsto alcun addebito per la creazione di un bucket. Vengono addebitati solo i costi per lo storage degli oggetti nel bucket e per il trasferimento degli oggetti all'interno e all'esterno del bucket. I costi che vengono addebitati sulla base agli esempi riportati nella seguente guida sono minimi (meno di $1). Per ulteriori informazioni sui costi di storage, consulta Prezzi di Amazon S3.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

2. Nella barra di navigazione nella parte superiore della pagina, scegli il nome del file attualmente visualizzato Regione AWS. Quindi, scegli la regione in cui desideri creare un bucket.

   > **Note**
   >
   > Scegli una regione nelle tue vicinanze per ridurre al minimo la latenza e i costi o essere conforme ai requisiti normativi. Gli oggetti archiviati in una regione non la lasciano mai a meno che non vengano trasferiti esplicitamente in un'altra regione. Per un elenco di Amazon S3 Regioni AWS, consulta gli Servizio AWS endpoint in. Riferimenti generali di Amazon Web Services

3. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
4. Scegliere Create bucket (Crea bucket).

   Viene visualizzata la pagina Create bucket (Crea bucket).
5. In Configurazione generale, visualizza Regione AWS dove verrà creato il bucket.
6. In Tipo di bucket, scegli Scopo generale.
7. In Nome bucket, immettere il nome del bucket.

   Il nome del bucket deve:

   - Essere univoco all'interno di una partizione. Una partizione è un raggruppamento di regioni. AWS ha attualmente tre partizioni: `aws` (regioni standard), `aws-cn` (regioni Cina) e `aws-us-gov` (AWS GovCloud (US) Regions).
   - Deve contenere da 3 a 63 caratteri
   - Essere costituito solo da lettere minuscole, numeri, punti (.) e trattini (-). Per una migliore compatibilità, si consiglia di evitare l'utilizzo di punti (.) nei nomi dei bucket, ad eccezione dei bucket utilizzati solo per l'hosting di siti Web statici.
   - Iniziare e finire con una lettera o un numero.

   Una volta creato il bucket, non è possibile modificarne il nome. Per ulteriori informazioni sulla denominazione dei bucket, consulta Regole di denominazione dei bucket.

> **Important**
>
> Evitare di includere informazioni riservate, ad esempio numeri di account, nel nome del bucket. Il nome bucket è visibile nell'URL che punta agli oggetti nel bucket.

8. AWS Management Console ti consente di copiare le impostazioni di un bucket esistente nel tuo nuovo bucket. Se non desideri copiare le impostazioni di un bucket esistente, vai al passaggio successivo.

   > **Note**
   >
   > Questa opzione:
   >
   > - non è disponibile in AWS CLI ed è disponibile solo nella console
   > - Non è disponibile per i bucket di directory
   > - Non copia la policy del bucket dal bucket esistente al nuovo bucket

   Per copiare le impostazioni di un bucket esistente, in Copia le impostazioni dal bucket esistente, seleziona Scegli il bucket. Si apre la finestra Scegli il bucket. Trova il bucket con le impostazioni che desideri copiare e seleziona Scegli il bucket. La finestra Scegli il bucket si chiude e la finestra Crea bucket si riapre.

   In Copia le impostazioni dal bucket esistente, ora vedrai il nome del bucket selezionato. Vedrai anche l'opzione Ripristina i valori predefiniti che puoi usare per rimuovere le impostazioni del bucket copiato. Controlla le impostazioni rimanenti del bucket, nella pagina Crea bucket. Vedrai che ora corrispondono alle impostazioni del bucket che hai selezionato. Puoi passare alla fase finale.

9. Alla voce Proprietà oggetto, per disabilitare o abilitare le ACL e controllare la proprietà degli oggetti caricati nel bucket, scegliere una delle seguenti impostazioni:

   ACL disabilitate

   - Proprietario del bucket applicato (impostazione predefinita): le ACL sono disabilitate e il proprietario del bucket possiede automaticamente e ha il controllo completo di ogni oggetto nel bucket. Le liste di controllo degli accessi (ACL) non influiscono più sulle autorizzazioni di

accesso ai dati nel bucket S3. Il bucket utilizza le policy esclusivamente per definire il controllo degli accessi.

Per impostazione predefinita, le ACL sono disabilitate. La maggior parte degli attuali casi d'uso in Amazon S3 non richiede più l'uso delle ACL. È consigliabile mantenere le ACL disabilitate, tranne in circostanze insolite in cui è necessario controllare individualmente l'accesso per ciascun oggetto. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

ACL abilitate

- Proprietario del bucket scelto – Il proprietario del bucket possiede e ha il pieno controllo sui nuovi oggetti che altri account scrivono nel bucket con l'ACL predefinita `bucket-owner-full-control`.

  Se applichi l'impostazione Proprietario del bucket preferito, per richiedere che tutti i caricamenti di Amazon S3 includano l'ACL predefinita `bucket-owner-full-control`, puoi aggiungere una policy del bucket che consenta solo il caricamento di oggetti che utilizzano questa ACL.
- Scrittore di oggetti: chi carica un oggetto possiede l'oggetto, ne ha il pieno controllo e può concedere ad altri utenti l'accesso ad esso tramite ACL. Account AWS

> **Note**
>
> L'impostazione predefinita è Proprietario del bucket applicato. Per applicare l'impostazione predefinita e mantenere gli ACL disabilitati, è necessaria solo l'autorizzazione `s3:CreateBucket`. Per abilitare gli ACL, è necessario disporre dell'autorizzazione `s3:PutBucketOwnershipControls`.

10. In Impostazioni di blocco dell'accesso pubblico per questo bucket scegli le impostazioni di blocco dell'accesso pubblico che vuoi applicare al bucket.

    Per impostazione predefinita, tutte e quattro le impostazioni Blocco dell'accesso pubblico sono abilitate. È consigliabile mantenere tutte le impostazioni abilitate, a meno che non sia necessario disattivarne una o più di una per il caso d'uso specifico. Per ulteriori informazioni sul blocco dell'accesso pubblico, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

> **Note**
>
> Per abilitare tutte le impostazioni Blocco dell'accesso pubblico, è richiesta solo l'autorizzazione `s3:CreateBucket`. Per disattivare le impostazioni Blocco dell'accesso pubblico, è necessario disporre dell'autorizzazione `s3:PutBucketPublicAccessBlock`.

11. (Facoltativo) In Bucket Versioning (Controllo delle versioni bucket), puoi scegliere se conservare varianti degli oggetti nel bucket. Per ulteriori informazioni sulla funzione Controllo delle versioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

    Per disabilitare o abilitare il controllo delle versioni nel bucket, scegli Disable (Disabilita) o Enable (Abilita).

12. (Facoltativo) In Tags (Tag), puoi scegliere di aggiungere tag al bucket. I tag sono coppie chiave-valore utilizzate per classificare lo spazio di archiviazione.

    Per aggiungere un tag al bucket, inserisci un valore in Key (Chiave) e facoltativamente un valore in Value (Valore), quindi scegli Add Tag (Aggiungi tag).

13. In Default encryption (Crittografia di default), scegliere Edit (Modifica).

14. Per configurare la crittografia predefinita, in Tipo di crittografia scegli una delle seguenti opzioni:

    - Chiavi gestite Amazon S3 (SSE-S3)
    - AWS Key Management Service chiave (SSE-KMS)

    > **Important**
    >
    > Se usi l'opzione SSE-KMS per la configurazione della crittografia predefinita, sei soggetto alla quota delle richieste al secondo di AWS KMS. Per ulteriori informazioni sulle AWS KMS quote e su come richiedere un aumento delle quote, consulta Quotas nella Developer Guide.AWS Key Management Service

    I bucket e i nuovi oggetti sono crittografati con la crittografia lato server con una chiave gestita da Amazon S3 come livello base di configurazione della crittografia. Per ulteriori informazioni sulla crittografia predefinita, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3.

Per ulteriori informazioni sull'utilizzo della crittografia lato server di Amazon S3 per crittografare i dati, consulta Uso della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).

15. Se scegli Chiave AWS Key Management Service (SSE-KMS), procedi come segue:

    a. In Chiave AWS KMS specifica la tua chiave KMS in uno dei seguenti modi:

    - Per scegliere da un elenco di chiavi KMS disponibili, scegli tra le tue AWS KMS keys chiavi KMS e scegli la tua chiave KMS dall'elenco delle chiavi disponibili.

      In questo elenco vengono visualizzate sia la chiave Chiave gestita da AWS (`aws/s3`) che quella gestita dai clienti. Per ulteriori informazioni sulle chiavi gestite dal cliente, consulta Chiavi gestite dal cliente e chiavi AWS nella Guida per gli sviluppatori di AWS Key Management Service .

    - Per specificare l'ARN della chiave KMS, scegli Inserisci l'ARN della AWS KMS key e quindi specifica l'ARN della chiave KMS nel campo visualizzato.

    - Per creare una nuova chiave gestita dal cliente nella AWS KMS console, scegli Crea una chiave KMS.

      Per ulteriori informazioni sulla creazione di una AWS KMS key, consulta Creating keys nella AWS Key Management Service Developer Guide.

      Important

      Puoi utilizzare solo le chiavi KMS disponibili nello Regione AWS stesso bucket. La console Amazon S3 elenca solo le prime 100 chiavi KMS nella stessa regione del bucket. Per utilizzare una chiave KMS non elencata, devi inserire l'ARN della chiave KMS. Se desideri utilizzare una chiave KMS di proprietà di un account diverso, è necessario innanzitutto disporre dell'autorizzazione necessaria per l'uso della chiave e quindi inserire l'ARN della chiave KMS. Per ulteriori informazioni sulle autorizzazioni tra account per le chiavi KMS, consulta Creazione di chiavi KMS utilizzabili da altri account nella Guida per gli sviluppatori di AWS Key Management Service . Per ulteriori informazioni su SSE-KMS, consulta Specifica della crittografia lato server con AWS KMS (SSE-KMS).
      Quando utilizzi una AWS KMS key crittografia lato server in Amazon S3, devi scegliere una chiave KMS di crittografia simmetrica. Amazon S3 supporta solo chiavi KMS di crittografia simmetriche e non chiavi KMS asimmetriche. Per ulteriori

informazioni, consulta Identificazione delle chiavi KMS simmetriche e asimmetriche nella Guida per gli sviluppatori di AWS Key Management Service .

Per ulteriori informazioni sulla creazione di una AWS KMS key, consulta Creating keys nella Developer Guide.AWS Key Management Service Per ulteriori informazioni sull'utilizzo AWS KMS con Amazon S3, consulta. Utilizzo della crittografia lato server con chiavi (SSE-KMS) AWS KMS

b. Quando configuri il bucket per utilizzare la crittografia predefinita con SSE-KMS puoi anche abilitare le chiavi bucket S3. S3 Bucket Keys riduce il costo della crittografia diminuendo il traffico di richieste da Amazon S3 a. AWS KMS Per ulteriori informazioni, consulta Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3.

   Per utilizzare le chiavi bucket S3, in Chiave bucket seleziona Abilita.

16. (Facoltativo) Se si desidera abilitare il blocco oggetti S3, effettua le seguenti operazioni:

a. Scegli Impostazioni avanzate.

> **Important**
>
> L'abilitazione del blocco oggetti consente anche la funzione Controllo delle versioni del bucket. Dopo averlo abilitato, per il blocco di oggetti è necessario configurare le impostazioni predefinite di conservazione e di blocco di carattere legale per proteggere i nuovi oggetti dall'eliminazione o dalla sovrascrittura.

b. Se desideri abilitare il blocco degli oggetti, scegli Enable (Abilita), leggi l'avviso visualizzato e confermalo.

Per ulteriori informazioni, consulta Utilizzo del blocco oggetti S3.

> **Note**
>
> Per creare un bucket abilitato per il blocco degli oggetti, devi disporre delle seguenti autorizzazioni: `s3:CreateBucket`, `s3:PutBucketVersioning` e `s3:PutBucketObjectLockConfiguration`.

17. Scegliere Create bucket (Crea bucket).

È stato creato un bucket in Amazon S3.

Approfondimenti

Per aggiungere un oggetto al bucket, consulta Fase 2: Carica un oggetto nel tuo bucket.

Fase 2: Carica un oggetto nel tuo bucket

Dopo aver creato un bucket in Amazon S3, sei pronto per caricare un oggetto nel bucket. Un oggetto può essere qualsiasi tipo di file: file di testo, immagine, video e così via.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Per caricare un oggetto in un bucket

1. Apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket seleziona il nome del bucket in cui desideri caricare l'oggetto.
3. Nella scheda Oggetti del bucket seleziona Carica.
4. In File e cartelle, seleziona Aggiungi file.
5. Seleziona un file da caricare, quindi scegli Apri.
6. Scegli Carica.

Hai caricato correttamente un oggetto nel bucket.

Approfondimenti

Per visualizzare l'oggetto, consulta Fase 3: donwload di un oggetto.

Fase 3: donwload di un oggetto

Dopo avere caricato un oggetto in un bucket, è possibile visualizzare le informazioni sull'oggetto e scaricare l'oggetto nel computer locale.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Utilizzo della console S3

In questa sezione viene illustrato come utilizzare la console Amazon S3 per scaricare un oggetto da un bucket S3.

> **Note**
>
> - Puoi scaricare un solo oggetto alla volta.
> - Se utilizzi la console di Amazon S3 per scaricare un oggetto il cui nome della chiave termina con un punto (`.`), il punto viene rimosso dal nome della chiave dell'oggetto scaricato. Per mantenere il punto alla fine del nome dell'oggetto scaricato, devi utilizzare AWS Command Line Interface (AWS CLI), gli AWS SDK o REST API di Amazon S3.

Per scaricare un oggetto da un bucket S3

1. Accedi alla AWS Management Console e apri la console di Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegliere il nome del bucket dal quale si desidera scaricare un oggetto.
3. È possibile scaricare un oggetto da un bucket S3 in uno qualsiasi dei modi seguenti:
   - Seleziona la casella di controllo accanto all'oggetto e scegli Scarica. Se desideri scaricare l'oggetto in una cartella specifica, nel menu Azioni, scegli Scarica come.
   - Se desideri scaricare una versione specifica dell'oggetto, attiva Mostra versioni (che si trova accanto alla casella di ricerca). Seleziona la casella di controllo accanto alla versione dell'oggetto desiderato e scegli Scarica. Se desideri scaricare l'oggetto in una cartella specifica, nel menu Azioni, scegli Scarica come.

Hai scaricato correttamente il tuo oggetto.

Approfondimenti

Per copiare e incollare il tuo oggetto in Amazon S3, consulta Fase 4: copiare l'oggetto in una cartella.

Fase 4: copiare l'oggetto in una cartella

Hai aggiunto un oggetto a un bucket e hai scaricato l'oggetto. Ora, crei una cartella e copi l'oggetto, quindi lo incolli nella cartella.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Per copiare un oggetto in una cartella

1. Nell'elenco Buckets (Bucket), scegliere il nome del bucket.
2. Scegliere Create folder (Crea cartella) e configurare una nuova cartella:
   a. Immettere un nome di cartella (ad esempio, `favorite-pics`).
   b. Per le impostazioni di crittografia della cartella, scegliere Disable (Disabilita).
   c. Selezionare Salva.
3. Accedere al bucket o alla cartella Amazon S3 che contiene gli oggetti da copiare.
4. Selezionare la casella di controllo a sinistra dei nomi degli oggetti da copiare.
5. Scegliere Actions (Operazioni) e quindi Copy (Copia) nell'elenco di opzioni visualizzato.

   In alternativa, scegliere Copy (Copia) nelle opzioni in alto a destra.
6. Scegliere la cartella di destinazione:
   a. Seleziona Sfoglia S3.
   b. Scegliere il pulsante di opzione a sinistra del nome della cartella.

      Per passare a un'altra cartella e scegliere una sottocartella come destinazione, scegliere il nome della cartella.
   c. Scegliere Choose destination (Scegli destinazione).

Il percorso della cartella di destinazione viene visualizzato nella casella Destination (Destinazione). In alternativa, puoi immettere il percorso di destinazione nella cartella Destination (Destinazione), ad esempio s3://*nome-bucket*/*nome-cartella*/.

7. In basso a destra scegliere Copy (Copia).

   Amazon S3 copia gli oggetti nella cartella di destinazione.

Approfondimenti

Per eliminare un oggetto e un bucket in Amazon S3, consulta Fase 5: eliminare gli oggetti e il bucket.

Fase 5: eliminare gli oggetti e il bucket

Quando non hai più bisogno di un oggetto o di un bucket, ti consigliamo di eliminarlo per evitare ulteriori addebiti. Se hai completato questa procedura dettagliata iniziale come esercizio di apprendimento e non hai intenzione di utilizzare il bucket o gli oggetti, ti consigliamo di eliminare il bucket in modo che non si accumulino più addebiti.

Prima di eliminare il bucket, devi svuotare il bucket o eliminare gli oggetti nel bucket. Una volta eliminati gli oggetti e il bucket non sono più disponibili.

Se desideri continuare a utilizzare lo stesso nome di bucket, è consigliabile eliminare gli oggetti o svuotare il bucket senza eliminarlo. Dopo aver eliminato un bucket, il nome diventa disponibile per il riutilizzo. Tuttavia, un altro Account AWS potrebbe creare un bucket con lo stesso nome prima che tu possa riutilizzarlo.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Argomenti

- Eliminazione di un oggetto
- Svuotamento del bucket

- Eliminazione del bucket

Eliminazione di un oggetto

Se desideri scegliere quali oggetti eliminare senza svuotare tutti gli oggetti dal bucket, puoi eliminare un oggetto.

1. Nell'elenco Buckets (Bucket) scegliere il nome del bucket dal quale si desidera eliminare un oggetto.
2. Seleziona l'oggetto da eliminare.
3. Scegli Elimina nelle opzioni disponibili in alto a destra.
4. Nella pagina Elimina oggetti, digita **`delete`** per confermare l'eliminazione degli oggetti.
5. Scegliere Delete objects (Elimina oggetti).

Svuotamento del bucket

Se pensi di eliminare il bucket, devi prima svuotarlo, eliminando così tutti gli oggetti nel bucket.

Per svuotare un bucket

1. Nell'elenco Buckets (Bucket) selezionare il bucket che si desidera svuotare e quindi scegliere Empty (Svuota).
2. Per confermare che si desidera svuotare il bucket ed eliminare tutti gli oggetti in esso contenuti, in Svuota bucket, digita **`permanently delete`**.

   > **Important**
   >
   > Lo svuotamento del bucket non può essere annullato. Gli oggetti aggiunti al bucket mentre l'azione di svuotamento del bucket è in corso verranno eliminati.

3. Per svuotare il bucket ed eliminare tutti gli oggetti in esso contenuti, scegliere Empty (Svuota).

   Viene visualizzata la pagina sullo stato dello svuotamento del bucket che è possibile utilizzare per esaminare un riepilogo delle eliminazioni di oggetti non riuscite e riuscite.
4. Per tornare all'elenco dei bucket, scegliere Exit (Esci).

Eliminazione del bucket

Dopo aver svuotato il bucket o eliminato tutti gli oggetti dal bucket, è possibile eliminarlo.

1. Per eliminare un bucket, nell'elenco Buckets (Bucket) selezionare il bucket.
2. Scegliere Delete (Elimina).
3. Per confermare l'eliminazione, in Elimina bucket, specifica il nome del bucket.

> **Important**
>
> L'eliminazione di un bucket non può essere annullata. I nomi dei bucket sono univoci. Se elimini il bucket, un altro utente AWS può utilizzare il nome. Se desideri continuare a utilizzare lo stesso nome di bucket, non eliminare il bucket. Invece, svuota il bucket e conservalo.

4. Per eliminare il bucket, scegliere Delete bucket (Elimina bucket).

Passaggi successivi

Negli esempi precedenti hai imparato a eseguire alcuni processi di base di Amazon S3.

I seguenti argomenti illustrano i percorsi di apprendimento che puoi sfruttare per acquisire una maggiore conoscenza di Amazon S3 in modo da implementarlo nelle tue applicazioni.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Argomenti

- Conoscere i casi d'uso comuni
- Controllo dell'accesso a bucket e oggetti
- Gestire e monitorare l'archiviazione
- Sviluppo con Amazon S3
- Informazioni sui tutorial

- Esplora la formazione e il supporto

Conoscere i casi d'uso comuni

Puoi utilizzare Amazon S3 per supportare il tuo caso d'uso specifico. La AWS Libreria di soluzioni e il Blog AWS forniscono informazioni e tutorial specifici per i casi d'uso. Di seguito sono elencati alcuni casi d'uso comuni per Amazon S3:

- Backup e archiviazione – Utilizza le caratteristiche di gestione dell'archiviazione di Amazon S3 per gestire i costi, soddisfare i requisiti normativi, ridurre la latenza e salvare più copie distinte dei dati per i requisiti di conformità.
- Hosting di applicazioni: distribusci, installa e gestisci applicazioni Web affidabili, altamente scalabili e a basso costo. Per esempio, è possibile configurare il bucket di Amazon S3 per l'hosting di siti Web statici. Per ulteriori informazioni, consulta Hosting di un sito Web statico tramite Amazon S3.
- Hosting di file multimediali: crea un'infrastruttura ad alta disponibilità per l'hosting di video, foto o per caricare e scaricare file musicali.
- Distribuzione di software: esegui l'hosting di applicazioni software che i clienti possono scaricare.

Controllo dell'accesso a bucket e oggetti

Amazon S3 offre una varietà di funzionalità e strumenti di sicurezza. Per una panoramica, consulta Gestione degli accessi.

Per impostazione predefinita, i bucket S3 e gli oggetti al loro interno sono privati. Puoi accedere solo alle risorse S3 che hai creato. Puoi utilizzare le seguenti caratteristiche per concedere autorizzazioni granulari delle risorse che supportano il tuo caso d'uso specifico o per verificare le autorizzazioni delle tue risorse Amazon S3.

- Blocco dell'accesso pubblico di S3: blocca l'accesso pubblico a bucket S3 e oggetti. Per impostazione predefinita, le impostazioni Blocco dell'accesso pubblico sono attivate a livello di bucket.
- AWS Identity and Access Management Identità (IAM): usa IAM o AWS IAM Identity Center crea identità IAM nel tuo sistema Account AWS per gestire l'accesso alle tue risorse Amazon S3. Ad esempio, puoi utilizzare IAM con Amazon S3 per controllare il tipo di accesso di un utente o di un gruppo di utenti a un bucket Amazon S3 di tua proprietà. Account AWS Per ulteriori informazioni

sulle identità IAM e sulle best practice, consulta Identità IAM (utenti, gruppi di utenti e ruoli) nella Guida per l'utente di IAM.

- Policy di bucket: utilizza il linguaggio delle policy basato su IAM per configurare le autorizzazioni basate sulle risorse per i bucket S3 e gli oggetti in essi contenuti.
- Liste di controllo degli accessi (ACL): concedi autorizzazioni di lettura e scrittura per singoli bucket e oggetti agli utenti autorizzati. Come regola generale, è consigliabile utilizzare policy basate sulle risorse S3 (policy di bucket e policy dei punti di accesso) o policy utente IAM per il controllo degli accessi anziché ACL. Le policy sono un'opzione di controllo degli accessi semplificata e più flessibile. Con le policy di bucket e le policy dei punti di accesso, è possibile definire regole valide globalmente per tutte le richieste alle risorse Amazon S3. Per ulteriori informazioni su casi specifici quando desideri utilizzare le ACL anziché le policy basate sulle risorse o le policy utente IAM, consultare Identity and Access Management per Amazon S3.
- S3 Proprietà dell'oggetto: consente di assumere la proprietà di ogni oggetto nel bucket, semplificando la gestione degli accessi per i dati archiviati in Amazon S3. S3 Proprietà dell'oggetto è un'impostazione a livello di bucket di Amazon S3 che puoi utilizzare per disabilitare o abilitare le ACL. Per impostazione predefinita, le ACL sono disabilitate. Con le ACL disabilitate, il proprietario del bucket possiede tutti gli oggetti nel bucket e gestisce l'accesso ai dati in maniera esclusiva utilizzando policy di gestione dell'accesso.
- IAM Access Analyzer per S3: valuta e monitora le policy di accesso al bucket S3, assicurando che forniscano solo l'accesso previsto alle risorse S3.

Gestire e monitorare l'archiviazione

- Gestione dell'archiviazione: dopo aver creato bucket e caricato oggetti in Amazon S3, puoi gestire l'archiviazione degli oggetti. Ad esempio, puoi utilizzare il controllo delle versioni S3 e la replica S3 per il ripristino di emergenza, il ciclo di vita S3 per gestire i costi di archiviazione e il blocco oggetti S3 per soddisfare i requisiti di conformità.
- Monitoraggio dello storage: il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di Amazon S3 e delle tue AWS soluzioni. Puoi monitorare l'attività e i costi di archiviazione. È consigliabile raccogliere i dati sul monitoraggio da tutte le parti della soluzione AWS per consentire un debug più facile di eventuali guasti in più punti.
- Analisi dei dati e informazioni dettagliate: puoi utilizzare le analisi dei dati e le informazioni dettagliate in Amazon S3 per comprendere, analizzare e ottimizzare l'utilizzo dell'archiviazione. Ad esempio, utilizza Amazon S3 Storage Lens per comprendere, analizzare e ottimizzare l'archiviazione. S3 Storage Lens fornisce oltre 29 parametri di utilizzo e attività e dashboard

interattivi per aggregare i dati per l'intera organizzazione, account specifici, regioni, bucket o prefissi. Utilizza l'analisi della classe di archiviazione per analizzare i modelli di accesso all'archiviazione e decidere quando è il momento di spostare i dati in una classe di archiviazione più conveniente.

Sviluppo con Amazon S3

Amazon S3 è un servizio REST. Puoi inviare richieste ad Amazon S3 utilizzando l'API REST o le librerie AWS SDK, che racchiudono l'API REST di Amazon S3 sottostante, semplificando le attività di programmazione. Puoi anche utilizzare AWS Command Line Interface (AWS CLI) per effettuare chiamate API Amazon S3. Per ulteriori informazioni, consulta Esecuzione di richieste.

L'API REST Amazon S3 è un'interfaccia HTTP per Amazon S3. Con l'API REST, uilizzi le richieste HTTP standard per creare, recuperare ed eliminare bucket e oggetti. Per utilizzare l'API REST, puoi servirti di qualunque kit di strumenti in grado di supportare HTTP. Puoi anche utilizzare un browser per recuperare gli oggetti, purché siano leggibili in modo anonimo. Per ulteriori informazioni, consulta Sviluppo con Amazon S3 utilizzando l'API REST.

Per ottenere aiuto nella creazione di applicazioni mediante il linguaggio preferito, puoi fare riferimento alle seguenti risorse.

AWS CLI

Puoi accedere alle caratteristiche di Amazon S3 utilizzando AWS CLI. Per scaricare e configurare, consulta. AWS CLISviluppo con Amazon S3 tramite la AWS CLI

AWS CLI Fornisce due livelli di comandi per accedere ad Amazon S3: comandi di alto livello (s3) e comandi a livello di API (s3api e s3control). I comandi S3 di livello alto semplificano le operazioni di uso frequente, ad esempio la creazione, la modifica e l'eliminazione di oggetti e bucket. I comandi s3api e s3control espongono l'accesso diretto a tutte le operazioni tramite API di Amazon S3, che puoi utilizzare per eseguire operazioni avanzate che potrebbero non essere possibili solo con i comandi di livello alto.

Per un elenco di AWS CLI comandi Amazon S3, consulta s3, s3api e s3control.

AWS SDK ed Explorer

Puoi utilizzare gli AWS SDK per sviluppare applicazioni con Amazon S3. Tali SDK AWS semplificano le attività di programmazione tramite il wrapping dell'API REST sottostante. I AWS Mobile SDK e

la libreria JavaScript Amplify sono disponibili anche per la creazione di applicazioni mobili e web connesse. AWS

Oltre agli AWS SDK, sono disponibili AWS Explorer per Visual Studio ed Eclipse for Java IDE. In questo caso, gli SDK e gli explorer sono raggruppati insieme come Toolkit. AWS

Per ulteriori informazioni, consulta Sviluppo con Amazon S3 utilizzando gli SDK AWS.

Librerie e codice di esempio

Il Centro Developer di AWS e il Catalogo dei codici di esempio di AWS contengono codici di esempio e librerie compilati appositamente per Amazon S3. È possibile utilizzare tali codici di esempio per comprendere le modalità di implementazione dell'API di Amazon S3. Puoi anche visualizzare l'Documentazione di riferimento delle API di Amazon Simple Storage Service per comprendere in dettaglio le operazioni API di Amazon S3.

Informazioni sui tutorial

Puoi iniziare con step-by-step i tutorial per saperne di più su Amazon S3. I tutorial presentati sono solo esempi con nomi di società e utenti fittizi destinati a essere usati in un ambiente di laboratorio. Il loro scopo è di fornire linee guida di carattere generico. Non devono essere utilizzati direttamente nell'ambiente di produzione, senza un'accurata opera di revisione e adattamento alle necessità esclusive del tuo ambiente lavorativo.

Nozioni di base

- Tutorial: Archiviazione e recupero di file con Amazon S3
- Tutorial: Nozioni di base su Piano intelligente S3
- Tutorial: Nozioni di base sull'utilizzo delle classi di archiviazione di Amazon S3 Glacier

Ottimizzazione dei costi di archiviazione

- Tutorial: Nozioni di base su Piano intelligente S3
- Tutorial: Nozioni di base sull'utilizzo delle classi di archiviazione di Amazon S3 Glacier
- Tutorial: Ottimizzazione dei costi e acquisizione di visibilità sull'utilizzo con S3 Storage Lens

Gestione dello storage

- Tutorial: Nozioni di base sui punti di accesso multi-regione di Amazon S3
- Tutorial: Replicating existing objects in your Amazon S3 buckets with S3 Batch Replication (Replica di oggetti esistenti nei bucket Amazon S3 con S3 Batch Replication)

Hosting di video e siti Web

- Tutorial: hosting di video in streaming su richiesta con Amazon S3, Amazon e CloudFront Amazon Route 53
- Esercitazione: configurazione di un sito Web statico su Amazon S3
- Tutorial: Configurazione di un sito Web statico utilizzando un dominio personalizzato registrato con Route 53

Elaborazione di dati

- Tutorial: trasformazione dei dati per l'applicazione con S3 Object Lambda
- Tutorial: rilevamento e oscuramento dei dati PII con S3 Object Lambda e Amazon Comprehend
- Tutorial: utilizzo di S3 Object Lambda per aggiungere filigrane alle immagini in modo dinamico man mano che vengono recuperate
- Tutorial: transcodifica in batch di video con S3 Batch Operations e AWS LambdaAWS Elemental MediaConvert

Protezione dei dati

- Tutorial: Verifica dell'integrità dei dati in Amazon S3 con checksum aggiuntivi
- Tutorial: Replica dei dati all'interno e tra di essi utilizzando S3 Replication Regioni AWS
- Tutorial: Protezione dei dati su Amazon S3 da eliminazioni accidentali o bug delle applicazioni mediante le funzionalità S3 di controllo delle versioni, blocco degli oggetti e replica
- Tutorial: Replicating existing objects in your Amazon S3 buckets with S3 Batch Replication (Replica di oggetti esistenti nei bucket Amazon S3 con S3 Batch Replication)

Esplora la formazione e il supporto

Puoi imparare dagli AWS esperti per migliorare le tue competenze e ottenere l'assistenza degli esperti per raggiungere i tuoi obiettivi.

- Formazione: le risorse per la formazione offrono un approccio pratico all'apprendimento di Amazon S3. Per ulteriori informazioni, consulta AWS Training and Certification e colloqui tecnologici online di AWS.
- Forum di discussione: nel forum, puoi rivedere i post per capire quali sono le operazioni supportate da Amazon S3. Puoi anche pubblicare domande. Per ulteriori informazioni, consulta Forum di discussione.
- Supporto tecnico: in caso di ulteriori domande, puoi contattare il Supporto tecnico.

Tutorial

I seguenti tutorial presentano end-to-end procedure complete per le attività più comuni di Amazon S3. I tutorial presentati sono solo esempi con nomi di società e utenti fittizi destinati a essere usati in un ambiente di laboratorio. Il loro scopo è di fornire linee guida di carattere generico. Non devono essere utilizzati direttamente nell'ambiente di produzione, senza un'accurata opera di revisione e adattamento alle necessità esclusive del tuo ambiente lavorativo.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Nozioni di base

- Tutorial: Archiviazione e recupero di file con Amazon S3
- Tutorial: Nozioni di base su Piano intelligente S3
- Tutorial: Nozioni di base sull'utilizzo delle classi di archiviazione di Amazon S3 Glacier

Ottimizzazione dei costi di archiviazione

- Tutorial: Nozioni di base su Piano intelligente S3
- Tutorial: Nozioni di base sull'utilizzo delle classi di archiviazione di Amazon S3 Glacier
- Tutorial: Ottimizzazione dei costi e acquisizione di visibilità sull'utilizzo con S3 Storage Lens

Gestione dello storage

- Tutorial: Nozioni di base sui punti di accesso multi-regione di Amazon S3
- Tutorial: Replicating existing objects in your Amazon S3 buckets with S3 Batch Replication (Replica di oggetti esistenti nei bucket Amazon S3 con S3 Batch Replication)

Hosting di video e siti Web

- Tutorial: hosting di video in streaming su richiesta con Amazon S3, Amazon e CloudFront Amazon Route 53
- Esercitazione: configurazione di un sito Web statico su Amazon S3
- Tutorial: Configurazione di un sito Web statico utilizzando un dominio personalizzato registrato con Route 53

Elaborazione di dati

- Tutorial: trasformazione dei dati per l'applicazione con S3 Object Lambda
- Tutorial: rilevamento e oscuramento dei dati PII con S3 Object Lambda e Amazon Comprehend
- Tutorial: utilizzo di S3 Object Lambda per aggiungere filigrane alle immagini in modo dinamico man mano che vengono recuperate
- Tutorial: transcodifica in batch di video con S3 Batch Operations e AWS LambdaAWS Elemental MediaConvert

Protezione dei dati

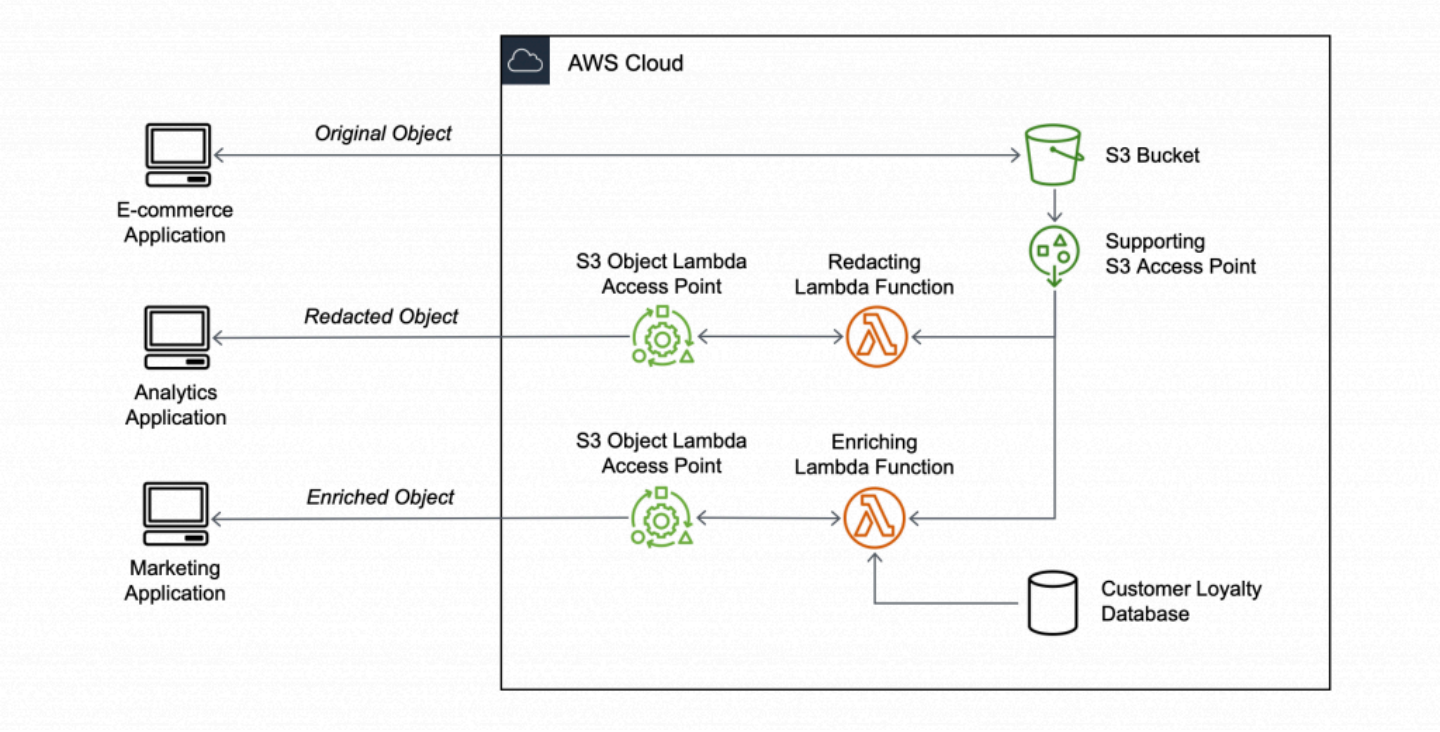
- Tutorial: Verifica dell'integrità dei dati in Amazon S3 con checksum aggiuntivi
- Tutorial: Replica dei dati all'interno e tra di essi utilizzando S3 Replication Regioni AWS
- Tutorial: Protezione dei dati su Amazon S3 da eliminazioni accidentali o bug delle applicazioni mediante le funzionalità S3 di controllo delle versioni, blocco degli oggetti e replica
- Tutorial: Replicating existing objects in your Amazon S3 buckets with S3 Batch Replication (Replica di oggetti esistenti nei bucket Amazon S3 con S3 Batch Replication)

Tutorial: trasformazione dei dati per l'applicazione con S3 Object Lambda

Quando archivi dati in Amazon S3, puoi condividerli facilmente per utilizzarli da più applicazioni. Tuttavia, ogni applicazione potrebbe avere requisiti univoci in merito al formato dei dati e richiedere la modifica o l'elaborazione dei dati per un caso d'uso specifico. Ad esempio, un set di dati creato

da un'applicazione e-commerce potrebbe includere informazioni personali di identificazione (PII). Quando gli stessi dati vengono elaborati per l'analisi, queste PII non sono necessarie e devono essere oscurate. Tuttavia, se lo stesso set di dati viene utilizzato per una campagna di marketing, potresti dover arricchire i dati con ulteriori dettagli, come informazioni dal database che raccoglie i dati sulla fidelizzazione dei clienti.

Con S3 Object Lambda puoi aggiungere il tuo codice per elaborare i dati recuperati da S3 prima di restituirli a un'applicazione. In particolare, puoi configurare una AWS Lambda funzione e collegarla a un S3 Object Lambda Access Point. Quando un'applicazione invia richieste GET S3 standard tramite il punto di accesso Lambda per oggetti S3, la funzione Lambda specificata viene richiamata per elaborare tutti i dati recuperati da un bucket S3 attraverso il punto di accesso S3 di supporto, quindi il punto di accesso Lambda per oggetti S3 restituisce il risultato trasformato all'applicazione. Puoi creare ed eseguire funzioni Lambda personalizzate, adattando la trasformazione dei dati S3 Object Lambda al tuo specifico caso d'uso, il tutto senza che siano necessarie modifiche alla tua applicazione.



Obiettivo

In questo tutorial imparerai come aggiungere codice personalizzato alle richieste GET S3 standard per modificare l'oggetto richiesto recuperato da S3 in modo che questo soddisfi le esigenze del client o dell'applicazione richiedente. In particolare, imparerai a trasformare in maiuscolo tutto il testo dell'oggetto originale archiviato in S3 attraverso S3 Object Lambda.

Argomenti

Prerequisiti

Prima di iniziare questo tutorial, devi disporre di un account a Account AWS cui accedere come utente AWS Identity and Access Management (IAM) con le autorizzazioni corrette. Devi inoltre installare la versione 3.8 o successiva di Python.

Fasi secondarie

- Creazione di un utente IAM con autorizzazioni nell' Account AWS (console)
- Installazione di Python 3.8 o versioni successive sul computer locale

Creazione di un utente IAM con autorizzazioni nell' Account AWS (console)

Puoi creare un utente IAM per il tutorial. Per completare questo tutorial, l'utente IAM deve allegare le seguenti policy IAM per accedere alle AWS risorse pertinenti ed eseguire azioni specifiche. Per ulteriori informazioni su come creare un utente IAM, consulta Creazione di utenti IAM (console) nella Guida per l'utente di IAM.

L'utente IAM richiede le seguenti policy:

- AmazonS3 FullAccess: concede le autorizzazioni per tutte le azioni di Amazon S3, incluse le autorizzazioni per creare e utilizzare un punto di accesso Object Lambda.
- AWSLambda_FullAccess— Concede le autorizzazioni per tutte le azioni Lambda.

- IAM FullAccess: concede le autorizzazioni a tutte le azioni IAM.
- IAM AccessAnalyzerReadOnlyAccess: concede le autorizzazioni per leggere tutte le informazioni di accesso fornite da IAM Access Analyzer.
- CloudWatchLogsFullAccess— Garantisce l'accesso completo ai log. CloudWatch

> **Note**
>
> Per semplicità, questo tutorial crea e utilizza un utente IAM. Dopo aver completato il tutorial, ricordati di Eliminazione dell'utente IAM. Per l'uso in produzione, consigliamo di seguire le best practice di sicurezza in IAM disponibili nella Guida per l'utente di IAM. Come best practice, richiedi agli utenti di utilizzare la federazione con un gestore di identità per accedere a AWS utilizzando credenziali temporanee. Un'ulteriore suggerimento derivante dalle best practice è richiedere ai carichi di lavoro di utilizzare credenziali temporanee con ruoli IAM per l'accesso ad AWS. Per informazioni sull'utilizzo AWS IAM Identity Center per creare utenti con credenziali temporanee, consulta Guida introduttiva nella Guida per l'AWS IAM Identity Center utente.
>
> Per semplicità, questo tutorial utilizza policy gestite AWS di accesso completo. Per l'utilizzo in produzione, è consigliabile invece concedere solo le autorizzazioni minime necessarie per il caso d'uso, in conformità con le best practice in fatto di sicurezza.

Installazione di Python 3.8 o versioni successive sul computer locale

Utilizza la procedura seguente per installare Python 3.8 o una versione successiva sul computer locale. Per maggiori istruzioni sull'installazione, consulta la pagina Download di Python nella Guida per principianti di Python.

1. Apri il terminale o la shell locale ed esegui il seguente comando per determinare se Python è già installato e, in caso affermativo, la versione installata.

   ```
   python --version
   ```

2. Se non disponi di Python 3.8 o versioni successive, scarica il programma di installazione ufficiale di Python 3.8 o versioni successive adatto al computer locale.
3. Esegui il programma di installazione facendo doppio clic sul file scaricato e segui i passaggi per completare l'installazione.

Per utenti Windows: scegli Add Python 3.X to PATH (Aggiungi Python 3.X a [percorso]) nella procedura guidata di installazione prima di scegliere Install Now (Installa adesso).

4. Chiudi e riapri il terminale per riavviarlo.
5. Per verificare che Python 3.8 o versioni successive sia installato correttamente, esegui il comando seguente.

   Se sei un utente macOS, esegui questo comando:

   ```
   python3 --version
   ```

   Per utenti Windows: esegui questo comando:

   ```
   python --version
   ```

6. Esegui il seguente comando per verificare che il gestore di pacchetti pip3 sia installato. Se vedi un numero di versione di pip e Python 3.8 o versione successiva nella risposta al comando, significa che il gestore di pacchetti pip3 è installato correttamente.

   ```
   pip --version
   ```

Fase 1: Creazione di un bucket S3

Crea un bucket per archiviare i dati originali che intendi trasformare.

Per creare un bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Scegliere Create bucket (Crea bucket).

   Viene visualizzata la pagina Create bucket (Crea bucket).
4. In Bucket name (Nome bucket), inserisci un nome per il bucket (ad esempio **`tutorial-bucket`**).

Per ulteriori informazioni sulle regole di denominazione del bucket in Amazon S3, consulta Regole di denominazione dei bucket.

5. Per Regione, scegli Regione AWS dove vuoi che risieda il bucket.

   Per ulteriori informazioni sulla regione del bucket, consulta Panoramica dei bucket.

6. In Block Public Access settings for this bucket (Blocca le impostazioni di accesso pubblico per questo bucket), mantieni le impostazioni predefinite (è abilitato Block all public access (Blocca tutto l'accesso pubblico).

   È consigliabile di lasciare abilitate tutte le impostazioni di blocco dell'accesso pubblico, a meno che non abbia bisogno di disattivarne una o più per il caso d'uso. Per ulteriori informazioni sul blocco dell'accesso pubblico, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

7. Mantieni le impostazioni rimanenti impostate sui valori di default.

   (Facoltativo) Se desideri configurare ulteriori impostazioni del bucket per il tuo caso d'uso specifico, consulta Creazione di un bucket.

8. Seleziona Crea bucket.

Fase 2: Caricamento di un file nel bucket S3

Carica il file di testo nel bucket S3. Questo file di testo contiene i dati originali che trasformerai in maiuscolo più avanti in questo tutorial.

Ad esempio, puoi caricare un file `tutorial.txt` che contiene il testo seguente:

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

Per caricare un file in un bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckets (Bucket) scegli il nome del bucket creato nella Fase 1 (ad esempio,**`tutorial-bucket`**) in cui caricare il file.

4. Nella scheda Oggetti del bucket seleziona Carica.
5. Nella pagina Upload (Caricamento), sotto Files and Folders (File e cartelle) scegli Add Files (Aggiungi file).
6. Seleziona un file da caricare, quindi scegli Apri. Ad esempio, puoi caricare il file di esempio `tutorial.txt` menzionato in precedenza.
7. Scegli Carica.

Fase 3: Creazione di un punto di accesso S3

Per utilizzare un punto di accesso Lambda per oggetti S3 per accedere e trasformare i dati originali, devi creare un punto di accesso S3 e associarlo al bucket S3 creato nella Fase 1. Il punto di accesso deve coincidere con Regione AWS gli oggetti che desideri trasformare.

Più avanti in questo tutorial, utilizzerai questo punto di accesso come punto di accesso di supporto per il tuo punto di accesso Lambda per oggetti.

Per creare un punto di accesso

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Access Points (Punti di accesso).
3. Nella pagina Access Points (Punti di accesso) scegli Create access point (Crea punto di accesso).
4. Nel campo Access point name (Nome del punto di accesso), inserisci il nome (per esempio, **`tutorial-access-point`**) per il punto di accesso.

   Per ulteriori informazioni sui punti di accesso S3, consulta Regole per la denominazione degli Punti di accesso Amazon S3.
5. Nel campo Bucket name (Nome bucket) inserisci il nome del bucket creato nella Fase 1 (ad esempio, **`tutorial-bucket`**). S3 allega quindi il punto di accesso a questo bucket.

   (Facoltativo) Puoi scegliere Browse S3 (Sfoglia S3) per sfogliare e cercare i bucket nell'account. Se scegli Browse S3 (Sfoglia S3), scegli il bucket desiderato e scegli Choose path (Scegli percorso) per popolare il campo Bucket name (Nome bucket) con il nome del bucket.
6. In Network origin (Origine rete), scegli Internet.

Per ulteriori informazioni sulle origini di rete per i punti di accesso, consulta Creazione di access point limitati a un cloud privato virtuale.

7. Tutte le impostazioni di blocco dell'accesso pubblico sono abilitate per impostazione predefinita per il punto di accesso. È consigliabile lasciare abilitato Block all public access (Blocca tutto l'accesso pubblico).

   Per ulteriori informazioni, consulta Gestione dell'accesso pubblico agli access point.

8. Per tutte le altre impostazioni del punto di accesso, mantieni i valori di default.

   (Facoltativo) Puoi modificare le impostazioni del punto di accesso per supportare il caso d'uso. Per questo tutorial, ti consigliamo di mantenere le impostazioni di default.

   (Facoltativo) Se è necessario gestire l'accesso al punto di accesso, puoi specificare una policy per il punto di accesso. Per ulteriori informazioni, consulta Esempi di policy degli access point.

9. Selezionare Crea punto di accesso.

Fase 4: Creazione di una funzione Lambda

Per trasformare i dati originali, crea una funzione Lambda utilizzabile con il punto di accesso Lambda per oggetti S3.

Fasi secondarie

- Scrittura del codice della funzione Lambda e creazione di un pacchetto di implementazione in un ambiente virtuale
- Creare una funzione Lambda con un ruolo di esecuzione (console)
- Implementa il codice della tua funzione Lambda con gli archivi in file .zip e configura la funzione Lambda (console)

Scrittura del codice della funzione Lambda e creazione di un pacchetto di implementazione in un ambiente virtuale

1. Sul computer locale, crea una cartella denominata `object-lambda` per l'ambiente virtuale da utilizzare più avanti in questo tutorial.

2. Nella cartella `object-lambda`, crea un file con una funzione Lambda che modifica tutto il testo dell'oggetto originale in maiuscolo. Ad esempio, puoi utilizzare la seguente funzione scritta in Python. Salva questa funzione in un file denominato `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 precedente funzione Lambda di esempio carica l'intero oggetto richiesto in memoria prima di trasformarlo e restituirlo al client. In alternativa, puoi trasmettere l'oggetto da

S3 per evitare di caricare l'intero oggetto in memoria. Questo approccio può essere utile quando lavori con oggetti di grandi dimensioni. Per ulteriori informazioni sullo streaming delle risposte con i punti di accesso Lambda per oggetti, consulta gli esempi di streaming in Utilizzo di richieste `GetObject` in Lambda.

Quando scrivi una funzione Lambda utilizzabile con un punto di accesso Lambda per oggetti S3, la funzione si basa sul contesto dell'evento di input che Lambda per oggetti S3 fornisce alla funzione stessa. Il contesto dell'evento fornisce informazioni relative alla richiesta eseguita nell'evento inviato da S3 Object Lambda a Lambda. e che contiene i parametri utilizzati per crearla.

I campi utilizzati per creare la funzione Lambda precedente sono i seguenti:

Il campo `getObjectContext` si riferisce ai dettagli di ingresso e uscita per le connessioni ad Amazon S3 e S3 Object Lambda. Ha i seguenti campi:

- `inputS3Url`: un URL prefirmato che la funzione Lambda può utilizzare per scaricare l'oggetto originale dal punto di accesso di supporto. Utilizzando un URL prefirmato, la funzione Lambda non ha bisogno di avere le autorizzazioni di lettura di Amazon S3 per recuperare l'oggetto originale e può accedere solo all'oggetto elaborato da ogni chiamata.
- `outputRoute`: un token di routing che viene aggiunto all'URL di S3 Object Lambda quando la funzione Lambda richiama `WriteGetObjectResponse` per reinviare l'oggetto trasformato.
- `outputToken`: un token utilizzato da S3 Object Lambda per associare la chiamata `WriteGetObjectResponse` al chiamante originale quando reinvia l'oggetto trasformato.

Per ulteriori informazioni su tutti i campi del contesto dell'evento, consulta Formato e utilizzo del contesto degli eventi e Scrittura di funzioni Lambda per i punti di accesso Lambda per oggetti S3.

3. Nel terminale locale, inserisci il seguente comando per installare il pacchetto `virtualenv`:

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

4. Nel terminale locale, apri la cartella `object-lambda` creata in precedenza e inserisci il seguente comando per creare e inizializzare un nuovo ambiente virtuale denominato `venv`.

```
python -m virtualenv venv
```

5. Per attivare l'ambiente virtuale, inserire il seguente comando per eseguire il file `activate` dalla cartella dell'ambiente:

   Se sei un utente macOS, esegui questo comando:

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

   Per utenti Windows: esegui questo comando:

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

   Il prompt dei comandi si modifica per mostrare (venv), indicando che l'ambiente virtuale è attivo.

6. Per installare le librerie richieste, esegui i seguenti comandi riga per riga nell'ambiente virtuale `venv`.

   Questi comandi installano versioni aggiornate delle dipendenze della funzione Lambda `lambda_handler`. Queste dipendenze sono gli SDK per Python (Boto3) AWS e il modulo delle richieste.

   ```
   pip3 install boto3
   ```

   ```
   pip3 install requests
   ```

7. Per disattivare l'ambiente virtuale, esegui il comando seguente:

   ```
   deactivate
   ```

8. Per creare un pacchetto di implementazione con le librerie installate come file `.zip` denominato `lambda.zip` alla root della directory `object-lambda`, eseguire i seguenti comandi riga per riga nel terminale locale.

   > **Tip**
   >
   > È possibile che i seguenti comandi debbano essere regolati per funzionare in un ambiente specifico. Ad esempio, una libreria potrebbe essere visualizzata in `site-packages` o `dist-packages` e la prima cartella potrebbe essere `lib` o `lib64`. Inoltre,

la cartella `python` potrebbe essere denominata con una versione di Python diversa. Per localizzare un pacchetto specifico, utilizza il comando `pip show`.

Se sei un utente macOS, esegui questi comandi:

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

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

Se sei un utente Windows, esegui questi comandi:

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

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

L'ultimo comando salva il pacchetto di implementazione nella directory principale della directory `object-lambda`.

9. Aggiungere il file del codice della funzione `transform.py` alla root del pacchetto di implementazione.

   Se sei un utente macOS, esegui questi comandi:

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

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

   Se sei un utente Windows, esegui questi comandi:

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

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

   Dopo aver completato questa fase, si dovrà avere la seguente struttura della directory:

   ```
   lambda.zip$
   ```

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

Creare una funzione Lambda con un ruolo di esecuzione (console)

1. Accedi AWS Management Console e apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/.

2. Nel pannello di navigazione a sinistra, scegli Functions (Funzioni).
3. Selezionare Create function (Crea funzione).
4. Scegli Author from scratch (Crea da zero).
5. In Basic information (Informazioni di base) eseguire queste operazioni:

   a. Nel campo Function name (Nome funzione), immettere **`tutorial-object-lambda-function`**.

   b. In Runtime (Runtime), scegli Python 3.8 o una versione successiva.
6. Espandi la sezione Change default execution role (Cambia ruolo di esecuzione predefinito). In Execution role (Ruolo di esecuzione), scegli Create a new role with basic Lambda permissions (Crea un nuovo ruolo con le autorizzazioni Lambda di base).

   Nel passaggio 5 più avanti di questo tutorial, colleghi AmazonS3 al ruolo di esecuzione di ObjectLambdaExecutionRolePolicy questa funzione Lambda.
7. Mantieni le impostazioni rimanenti impostate sui valori predefiniti.
8. Scegli Crea funzione.

Implementa il codice della tua funzione Lambda con gli archivi in file .zip e configura la funzione Lambda (console)

1. Nella AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/, scegli Funzioni nel riquadro di navigazione a sinistra.

2. Scegli la funzione Lambda creata in precedenza (ad esempio, **`tutorial-object-lambda-function`**).
3. Nella pagina dei dettagli della funzione Lambda, scegli la scheda Code (Codice). Nella sezione Code source (Origine codice), scegli Upload from (Carica da) e quindi .zip file (file .zip).
4. Scegli Upload (Carica) per selezionare il file `.zip` locale.
5. Scegli il file `lambda.zip` creato in precedenza, quindi scegli Open (Apri).
6. Selezionare Salva.
7. Nel pannello Runtime settings (Impostazioni runtime), scegli Edit (Modifica).
8. Nella pagina Edit runtime settings (Modifica impostazioni runtime), verifica che Runtime (Runtime) sia impostato su Python 3.8 o una versione successiva.
9. Per indicare al runtime Lambda quale metodo gestore richiamare nel codice della funzione Lambda, inserisci **`transform.lambda_handler`** in Handler (Gestore).

   Quando si configura una funzione in Python, il valore dell'impostazione del gestore è costituito dal nome del file e dal nome del modulo del gestore esportato, separati da un punto. Ad esempio, `transform.lambda_handler` richiama il metodo `lambda_handler` definito nel file `transform.py`.
10. Selezionare Salva.
11. (Facoltativo) Nella pagina dei dettagli della funzione Lambda, scegli la scheda Configuration (Configurazione). Nel pannello di navigazione a sinistra, scegli General configuration (Configurazione generale), quindi scegli Edit (Modifica). Nel campo Timeout inserisci **`1`** min **`0`** sec. Mantieni le impostazioni rimanenti sui valori predefiniti e scegli Save (Salva).

    Il Timeout è la quantità di tempo consentita da Lambda per l'esecuzione di una funzione per una chiamata prima di arrestarla. Il valore predefinito è 3 secondi. La durata massima per una funzione Lambda utilizzata da S3 Object Lambda è di 60 secondi. I prezzi si basano sulla quantità di memoria configurata e sulla quantità di tempo di esecuzione del codice.

Fase 5: Configurazione di una policy IAM per il ruolo di esecuzione della funzione Lambda

Per abilitare la funzione Lambda a fornire dati personalizzati e intestazioni di risposta al chiamante `GetObject`, il ruolo di esecuzione della funzione Lambda deve disporre delle autorizzazioni IAM per chiamare l'API `WriteGetObjectResponse`.

Allegare una policy IAM al ruolo della funzione Lambda

1. Nella AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/, scegli Funzioni nel riquadro di navigazione a sinistra.
2. Scegli la funzione creata nella Fase 4 (ad esempio, **`tutorial-object-lambda-function`**).
3. Nella pagina dei dettagli della funzione Lambda, scegli la scheda Configuration (Configurazione), quindi scegli Permissions (Autorizzazioni) nel pannello di navigazione a sinistra.
4. In Execution role (Ruolo di esecuzione) scegli il collegamento Role Name (Nome ruolo). Si apre la console IAM.
5. Nella pagina Summary (Riepilogo) della console IAM del ruolo di esecuzione della funzione Lambda, seleziona la scheda Permissions (Autorizzazioni). Quindi, nel menu Add Permissions (Aggiungi autorizzazioni), scegli Attach policies (Collega policy).
6. Nella pagina Attach permissions (Allega autorizzazioni) inserisci **`AmazonS3ObjectLambdaExecutionRolePolicy`** nella casella di ricerca per filtrare l'elenco di policy. Seleziona la casella di controllo accanto al nome della politica AmazonS3 ObjectLambdaExecutionRolePolicy.
7. Scegli Collega policy.

Fase 6: Creazione di un punto di accesso Lambda per oggetti S3

Un punto di accesso Lambda per oggetti S3 offre la flessibilità di richiamare una funzione Lambda direttamente da una richiesta GET S3 in modo che la funzione possa elaborare i dati recuperati da un punto di accesso S3. Quando crei e configuri un punto di accesso Lambda per oggetti S3, devi specificare la funzione Lambda da richiamare e fornire il contesto dell'evento in formato JSON come parametri personalizzati utilizzabili da Lambda.

Per creare un punto di accesso Lambda per oggetti S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Object Lambda Access Points (Punti di accesso Object Lambda).
3. Nella pagina Object Lambda Access Points (Punti di accesso Object Lambda), scegli Create Object Lambda Access Point (Crea punto di accesso Object Lambda).

4. In Nome del punto di accesso per le espressioni Lambda dell'oggetto immetti il nome che desideri utilizzare per il punto di accesso Lambda per oggetti (per esempio, **`tutorial-object-lambda-accesspoint`**).
5. In Supporting Access Point (Access point di supporto), inserisci o seleziona il punto di accesso standard creato nella Fase 3 (ad esempio, **`tutorial-access-point`**), quindi scegli Choose supporting Access Point (Scegli punto di accesso di supporto).
6. Per le API S3, per recuperare gli oggetti dal bucket S3 per l'elaborazione della funzione Lambda, seleziona. GetObject
7. In Invoke Lambda function (Chiama una funzione Lambda) per questo tutorial puoi scegliere una delle due opzioni seguenti.
   - Scegli Choose from functions in your account (Scegli tra le funzioni nell'account), dopodiché scegli la funzione Lambda creata nella Fase 4 (ad esempio, **`tutorial-object-lambda-function`**) dall'elenco a discesa Lambda function (Funzione Lambda).
   - Scegli Enter ARN (Inserisci ARN), quindi inserisci l'Amazon Resource Name (ARN) della funzione Lambda creata nella Fase 4.
8. In Lambda function version (Versione delle funzioni Lambda), scegli $LATEST (l'ultima versione della funzione Lambda creata nella Fase 4).
9. (Facoltativo) Se hai bisogno che la tua funzione Lambda riconosca ed elabori le richieste GET con intestazioni con intervalli e numeri di parte, seleziona Lambda function supports requests using range (La funzione Lambda supporta le richieste che utilizzano l'intervallo) e Lambda function supports requests using part numbers (La funzione Lambda supporta le richieste che utilizzano numeri di parte). Altrimenti, deselezionare queste due caselle di controllo.

   Per ulteriori informazioni sull'utilizzo di intervalli o numeri di parte con S3 Object Lambda, consulta Utilizzo delle intestazioni Range e partNumber.
10. (Facoltativo) In Payload - optional (Payload - facoltativo), aggiungi il testo JSON per fornire alla tua funzione Lambda ulteriori informazioni.

    Un payload è un testo JSON opzionale che puoi fornire alla tua funzione Lambda come input per tutte le chiamate provenienti da uno specifico punto di accesso Lambda per oggetti S3. Per personalizzare il comportamento di più punti di accesso Lambda per oggetti che richiamano la stessa funzione Lambda, puoi configurare i payload con parametri diversi, estendendo così la flessibilità della funzione stessa.

    Per ulteriori informazioni sul payload, consulta Formato e utilizzo del contesto degli eventi.

11. (Facoltativo) In Parametri di richiesta - facoltativo, scegli Disabilita o Abilita per aggiungere il monitoraggio Amazon S3 al punto di accesso Lambda per oggetti. Le metriche delle richieste vengono fatturate alla tariffa standard di Amazon CloudWatch . Per ulteriori informazioni, consultare Prezzi di CloudWatch.
12. In Object Lambda Access Point policy - optional (Policy del punto di accesso Object Lambda - facoltativo mantieni l'impostazione di default.

    (Facoltativo) Puoi impostare una policy delle risorse. Questa policy delle risorse fornisce all'API `GetObject` l'autorizzazione per utilizzare il punto di accesso Lambda per oggetti specificato.
13. Mantieni le impostazioni rimanenti sui valori di default, quindi scegliCreate Object Lambda Access Point (Crea punto di accesso Object Lambda).

Fase 7: Visualizzazione dei dati trasformati

S3 Object Lambda è ora pronto a trasformare i tuoi dati per il tuo caso d'uso. In questo tutorial, S3 Object Lambda trasforma tutto il testo dell'oggetto in maiuscolo.

Fasi secondarie

- Visualizzazione dei dati trasformati nel punto di accesso Lambda per oggetti S3
- Esegui uno script Python per stampare i dati originali e trasformati

Visualizzazione dei dati trasformati nel punto di accesso Lambda per oggetti S3

Quando chiedi di recuperare un file tramite il punto di accesso Lambda per oggetti S3, esegui una chiamata API `GetObject` a Lambda per oggetti S3. S3 Object Lambda richiama la funzione Lambda per trasformare i dati, dopodiché restituisce i dati trasformati come risposta alla chiamata API `GetObject` S3 standard.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Object Lambda Access Points (Punti di accesso Object Lambda).
3. Nella pagina Punti di accesso Lambda dell'oggetto scegli il punto di accesso Lambda per oggetti S3 creato nella Fase 6 (ad esempio, **`tutorial-object-lambda-accesspoint`**).
4. Nella scheda Oggetti del punto di accesso Lambda per oggetti S3, seleziona il file con lo stesso nome (ad esempio, `tutorial.txt`) di quello che hai caricato nel bucket S3 nella Fase 2.

Questo file deve contenere tutti i dati trasformati.

5. Per visualizzare i dati trasformati, scegli Open (Apri) o Download (Scarica).

Esegui uno script Python per stampare i dati originali e trasformati

Puoi utilizzare S3 Object Lambda con le tue applicazioni esistenti. A tale scopo, aggiorna la configurazione dell'applicazione in modo da utilizzare l'ARN del nuovo punto di accesso Lambda per oggetti S3 creato nella Fase 6 per recuperare i dati da S3.

Il seguente script Python di esempio stampa sia i dati originali dal bucket S3 sia i dati trasformati dal punto di accesso Lambda per oggetti S3.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Object Lambda Access Points (Punti di accesso Object Lambda).
3. Nella pagina Punti di accesso Lambda dell'oggetto scegli il pulsante di opzione a sinistra del punto di accesso Lambda per oggetti S3 creato nella Fase 6 (ad esempio, **`tutorial-object-lambda-accesspoint`**).
4. Scegli Copy ARN (Copia ARN).
5. Salva l'ARN per utilizzarlo in un secondo momento.
6. Scrivi uno script Python sul tuo computer locale per stampare sia i dati originali (ad esempio, `tutorial.txt`) dal tuo bucket S3 sia i dati trasformati (ad esempio, `tutorial.txt`) dal punto di accesso Lambda per oggetti S3. Puoi utilizzare il seguente script di esempio.

```
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. Salva il tuo script Python con un nome personalizzato (ad esempio, `tutorial_print.py`) nella cartella (ad esempio, `object-lambda`) che hai creato nella Fase 4 sul computer locale.
8. Nel terminale locale, esegui il seguente comando dalla root della directory (ad esempio, `object-lambda`) che hai creato nella Fase 4.

```
python3 tutorial_print.py
```

Dovresti vedere sia i dati originali sia i dati trasformati (tutto il testo in maiuscolo) attraverso il terminale. Per esempio l'output visualizzato sarà simile al testo seguente.

```
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.
```

Fase 8: Pulizia

Se hai trasformato i dati attraverso S3 Object Lambda solo come esercizio di apprendimento, elimina le risorse AWS che hai allocato per non accumulare più addebiti.

Fasi secondarie

- Eliminazione del punto di accesso Lambda per oggetti
- Eliminazione del punto di accesso S3
- Eliminazione del ruolo di esecuzione per la funzione Lambda
- Eliminazione della funzione Lambda
- Eliminare il gruppo di CloudWatch log
- Eliminazione del file originale nel bucket S3 di origine
- Eliminazione del bucket S3 di origine
- Eliminazione dell'utente IAM

Eliminazione del punto di accesso Lambda per oggetti

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Object Lambda Access Points (Punti di accesso Object Lambda).
3. Nella pagina Punti di accesso Lambda dell'oggetto scegli il pulsante di opzione a sinistra del punto di accesso Lambda per oggetti S3 creato nella Fase 6 (ad esempio, **`tutorial-object-lambda-accesspoint`**).
4. Scegli Elimina.
5. Conferma di voler eliminare il punto di accesso Lambda per oggetti inserendone il nome nel campo di testo che viene visualizzato, quindi scegli Elimina.

Eliminazione del punto di accesso S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Access Points (Punti di accesso).
3. Passa al punto di accesso creato nella Fase 3 (ad esempio, **`tutorial-access-point`**), quindi scegli il pulsante di opzione accanto al nome del punto di accesso.
4. Scegli Elimina.
5. Conferma di voler eliminare il punto di accesso inserendone il nome nel campo di testo che viene visualizzato, quindi scegli Delete (Elimina).

Eliminazione del ruolo di esecuzione per la funzione Lambda

1. Accedi AWS Management Console e apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/.
2. Nel pannello di navigazione a sinistra, scegli Functions (Funzioni).
3. Scegli la funzione creata nella Fase 4 (ad esempio, **`tutorial-object-lambda-function`**).
4. Nella pagina dei dettagli della funzione Lambda, scegli la scheda Configuration (Configurazione), quindi scegli Permissions (Autorizzazioni) nel pannello di navigazione a sinistra.
5. In Execution role (Ruolo di esecuzione) scegli il collegamento Role Name (Nome ruolo). Si apre la console IAM.
6. Nella pagina Summary (Riepilogo) della console IAM del ruolo di esecuzione della funzione Lambda, scegli Delete role (Elimina ruolo).
7. Nella finestra di dialogo Delete role (Elimina ruolo), scegli Yes, Delete (Sì, elimina).

Eliminazione della funzione Lambda

1. Nella AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/, scegli Funzioni nel riquadro di navigazione a sinistra.
2. Seleziona la casella di controllo a sinistra del nome della funzione creata nella Fase 4 (ad esempio, **`tutorial-object-lambda-function`**).
3. Scegli Azioni, quindi Elimina.
4. Nella finestra di dialogo Delete function (Elimina funzione), scegli Delete (Elimina).

Eliminare il gruppo di CloudWatch log

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione a sinistra, scegli Log groups (Gruppi di registri).
3. Individua il gruppo di registri il cui nome termina con la funzione Lambda creata nella Fase 4(ad esempio, **`tutorial-object-lambda-function`**).
4. Seleziona la casella di controllo a sinistra del nome del gruppo di registri.
5. Scegli Actions (Operazioni), quindi scegli Delete log group(s) (Elimina gruppi di registri).
6. Nella finestra di dialogo Delete log group(s) (Elimina gruppo/i di log) scegli Delete (Elimina).

Eliminazione del file originale nel bucket S3 di origine

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket name (Nome bucket) scegli il nome del bucket su cui hai caricato il file originale nella Fase 2 (ad esempio, **`tutorial-bucket`**).
4. Seleziona la casella di controllo a sinistra del nome dell'oggetto da eliminare (ad esempio, `tutorial.txt`).
5. Scegli Elimina.
6. Nella pagina Delete objects (Elimina oggetti), nella sezione Permanently delete objects? (Eliminare definitivamente gli oggetti?) conferma che desideri eliminare questo oggetto inserendo **`permanently delete`** nella casella di testo.
7. Scegliere Delete objects (Elimina oggetti).

Eliminazione del bucket S3 di origine

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckets (Bucket) seleziona il pulsante di opzione accanto al nome del bucket creato nella Fase 1 (ad esempio, **`tutorial-bucket`**).
4. Scegli Elimina.
5. Nella pagina Delete bucket (Elimina bucket) conferma che desideri eliminare il bucket inserendone il nome nel campo di testo e quindi scegli Delete bucket (Elimina bucket).

Eliminazione dell'utente IAM

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra, scegli Users (Utenti), quindi selezionare la casella di controllo accanto al nome utente che desideri eliminare.
3. Nella parte superiore della pagina, scegli Delete (Elimina).

4. Nella casella di dialogo Delete ***user name***? (Eliminare nome utente?) inserisci il nome utente nel campo di inserimento del testo per confermare l'eliminazione dell'utente. Scegli Elimina.

Passaggi successivi

Dopo aver completato questo tutorial, puoi personalizzare la funzione Lambda per il tuo caso d'uso in modo da modificare i dati restituiti dalle richieste GET S3 standard.

Di seguito è riportato un elenco di casi d'uso comuni per S3 Object Lambda:

- Mascheramento dei dati sensibili per la sicurezza e la conformità.

  Per ulteriori informazioni, consulta Tutorial: rilevamento e oscuramento dei dati PII con S3 Object Lambda e Amazon Comprehend.
- Filtraggio di determinate righe di dati per fornire informazioni specifiche.
- Arricchimento dei dati con informazioni provenienti da altri servizi o database.
- Conversione tra formati di dati, come la conversione di XML in JSON per la compatibilità delle applicazioni.
- Compressione o decompressione dei file durante il download.
- Ridimensionamento delle immagini e creazione della filigrana.

  Per ulteriori informazioni, consulta Tutorial: utilizzo di S3 Object Lambda per aggiungere filigrane alle immagini in modo dinamico man mano che vengono recuperate.
- Implementazione di regole di autorizzazione personalizzate per accedere ai dati.

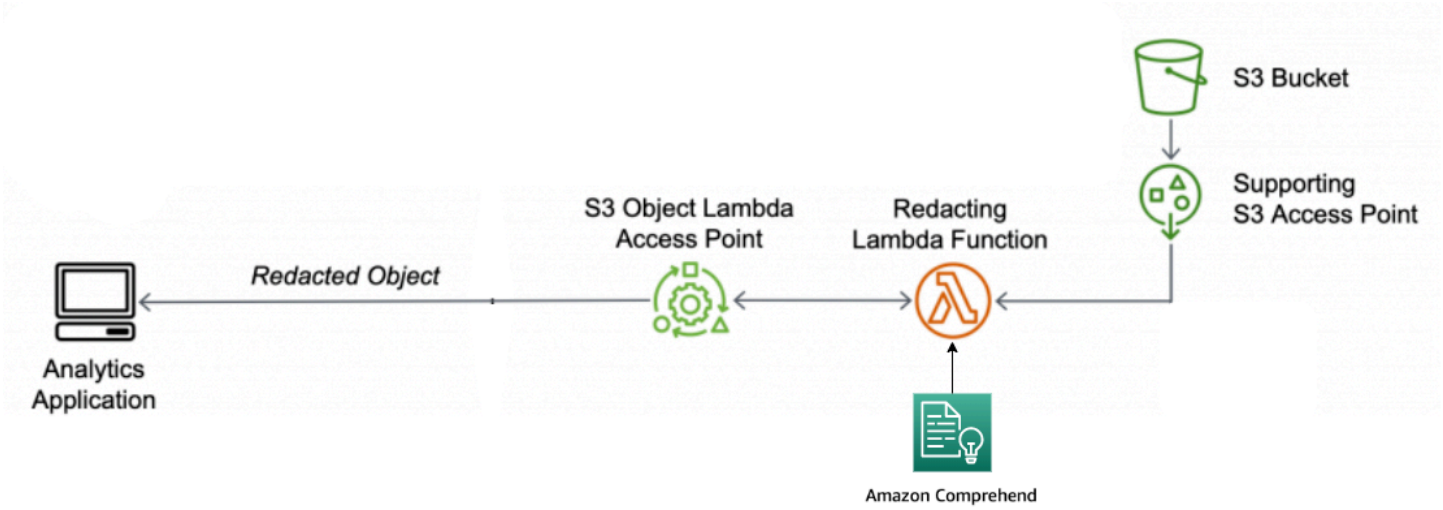
Per ulteriori informazioni su S3 Object Lambda, consulta Trasformazione di oggetti con S3 Object Lambda.

Tutorial: rilevamento e oscuramento dei dati PII con S3 Object Lambda e Amazon Comprehend

Quando utilizzi Amazon S3 per set di dati condivisi per l'accesso di più applicazioni e utenti, è importante limitare le informazioni privilegiate, ad esempio le informazioni personali di identificazione (PII), solo alle entità autorizzate. Ad esempio, quando un'applicazione di marketing utilizza alcuni dati contenenti PII, in primo luogo potrebbe essere necessario mascherare i dati PII per soddisfare

i requisiti di privacy dei dati. Inoltre, quando un'applicazione di analisi dei dati utilizza un set di dati di inventario con ordine di produzione, in primo luogo potrebbe essere necessario oscurare le informazioni della carta di credito del cliente per evitare perdite di dati non intenzionali.

Con S3 Object Lambda e una funzione AWS Lambda precostituita basata su Amazon Comprehend, puoi proteggere i dati PII recuperati da S3 prima di restituirli a un'applicazione. Nello specifico, puoi utilizzare la funzione Lambda precostituita come funzione di oscuramento e allegarla a un punto di accesso Lambda per oggetti S3. Quando un'applicazione (ad esempio, un'applicazione di analisi dei dati) invia richieste GET S3 standard, queste richieste effettuate tramite il punto di accesso Lambda per oggetti S3 richiamano la funzione Lambda precostituita di redazione per rilevare e oscurare i dati PII recuperati da un bucket S3 attraverso un punto di accesso S3 di supporto. Quindi, il punto di accesso Lambda per oggetti S3 restituisce il risultato oscurato all'applicazione.



Nel processo, la funzione Lambda precostituita utilizza Amazon Comprehend, un servizio di elaborazione del linguaggio naturale (NLP, Natural Language Processing), per acquisire variazioni nel modo in cui le PII sono rappresentate, indipendentemente dall'esistenza di PII nel testo (ad esempio in numeri o come combinazione di parole e numeri). Amazon Comprehend può anche utilizzare il contesto del testo per capire se un numero di 4 cifre è un PIN, gli ultimi quattro numeri di un numero di previdenza sociale (SSN) o un anno. Amazon Comprehend elabora qualsiasi file di testo in formato UTF-8 e può proteggere le PII su larga scala senza compromettere la precisione. Per ulteriori informazioni, consulta Cos'è Amazon Comprehend? nella Guida per Developer di Amazon Comprehend.

Obiettivo

In questo tutorial, imparerai a utilizzare S3 Object Lambda con la funzione Lambda precostituita `ComprehendPiiRedactionS3ObjectLambda`. Questa funzione utilizza Amazon Comprehend per

rilevare entità PII. Oscurare quindi queste entità sostituendole con asterischi. Oscurando le PII, si nascondono i dati sensibili, cosa che può contribuire alla sicurezza e alla conformità.

Imparerai anche a usare e configurare una AWS Lambda funzione predefinita per lavorare insieme AWS Serverless Application Repositorya S3 Object Lambda per una facile implementazione.

Argomenti

Prerequisiti: creazione di un utente IAM con autorizzazioni

Prima di iniziare questo tutorial, devi disporre di un AWS account a cui puoi accedere come AWS Identity and Access Management utente (utente IAM) con le autorizzazioni corrette.

Puoi creare un utente IAM per il tutorial. Per completare questo tutorial, l'utente IAM deve allegare le seguenti politiche IAM per accedere alle AWS risorse pertinenti ed eseguire azioni specifiche.

> **Note**
>
> Per semplicità, questo tutorial crea e utilizza un utente IAM. Dopo aver completato il tutorial, ricordati di Eliminazione dell'utente IAM. Per l'uso in produzione, consigliamo di seguire le best practice di sicurezza in IAM disponibili nella Guida per l'utente di IAM. Come best practice, richiedi agli utenti di utilizzare la federazione con un gestore di identità per accedere a AWS utilizzando credenziali temporanee. Un'ulteriore suggerimento derivante dalle best practice è richiedere ai carichi di lavoro di utilizzare credenziali temporanee con ruoli IAM per l'accesso ad AWS. Per ulteriori informazioni sull'utilizzo AWS IAM Identity Center per creare utenti con credenziali temporanee, consulta Guida introduttiva nella Guida per l'AWS IAM Identity Center utente.

> Per semplicità, questo tutorial utilizza policy di accesso completo. Per l'utilizzo in produzione, è consigliabile invece concedere solo le autorizzazioni minime necessarie per il caso d'uso, in conformità con le best practice in fatto di sicurezza.

Il tuo utente IAM richiede le seguenti politiche AWS gestite:

- AmazonS3 FullAccess: concede le autorizzazioni per tutte le azioni di Amazon S3, incluse le autorizzazioni per creare e utilizzare un punto di accesso Object Lambda.
- AWSLambda_FullAccess— Concede le autorizzazioni per tutte le azioni Lambda.
- AWSCloudFormationFullAccess— Concede le autorizzazioni per tutte le azioni. AWS CloudFormation
- IAM FullAccess: concede le autorizzazioni a tutte le azioni IAM.
- IAM AccessAnalyzerReadOnlyAccess: concede le autorizzazioni per leggere tutte le informazioni di accesso fornite da IAM Access Analyzer.

Puoi allegare direttamente queste policy esistenti durante la creazione di un utente IAM. Per ulteriori informazioni su come creare un utente IAM, consulta Creazione di utenti IAM (console) nella Guida per l'utente di IAM.

Inoltre, l'utente IAM richiede una policy gestita dal cliente. Per concedere all'utente IAM le autorizzazioni per tutte le AWS Serverless Application Repository risorse e le azioni, devi creare una policy IAM e allegarla all'utente IAM.

Per creare e allegare una policy IAM a un utente IAM

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra, seleziona Policy.
3. Scegli Crea policy.
4. Nella scheda Visual editor (Editor visivo), in Service (Servizio), seleziona Choose a Service (Scegli un servizio). Poi, scegli Serverless Application Repository.
5. In Actions (Operazioni), sotto Manual actions (Operazioni manuali), seleziona All Serverless Application Repository actions (serverlessrepo:*) (Tutte le operazioni Serverless Application Repository (serverlessrepo:*)) per questo tutorial.

Come best practice di sicurezza, dovresti concedere le autorizzazioni solo alle operazioni e alle risorse necessarie all'utente, in base al tuo caso d'uso. Per ulteriori informazioni, consulta Best Practice di sicurezza in IAM nella Guida per l'utente di IAM.

6. In Resources (Risorse), scegli All resources (Tutte le risorse) per questo tutorial.

   Come best practice, è consigliabile definire le autorizzazioni solo per risorse specifiche in account specifici. In alternativa, puoi concedere un privilegio minimo utilizzando le chiavi di condizione. Per ulteriori informazioni, consulta Assegnare il privilegio minimo nella Guida per l'utente di IAM.
7. Scegliere Next: Tags (Successivo: Tag).
8. Scegliere Next:Review (Successivo: Rivedi).
9. Nella pagina Review policy (Rivedi policy) digita i valori per Name (Nome) (per esempio, **`tutorial-serverless-application-repository`**) e Description (Descrizione) (facoltativa) per la policy che stai creando. Esaminare il riepilogo della policy per assicurarsi di aver concesso le autorizzazioni corrette e selezionare Create policy (Crea policy) per salvare la nuova policy.
10. Nel riquadro di navigazione a sinistra, seleziona Users (Utenti). Quindi, scegli l'utente IAM per questo tutorial.
11. Nella pagina Summary (Riepilogo) dell'utente scelto, scegli la scheda Permissions (Autorizzazioni), quindi scegli Add permissions (Aggiungi autorizzazioni).
12. In Grant permissions (Concedi autorizzazioni) scegli Attach existing policies directly (Collega direttamente le policy esistenti).
13. Seleziona la casella di controllo accanto alla policy creata (ad esempio, **`tutorial-serverless-application-repository`**) e quindi scegli Next: Review (Successivo: Rivedi).
14. In Permissions summary (Riepilogo delle autorizzazioni) esaminare il riepilogo per accertarti di aver allegato la policy desiderata. Quindi seleziona Add permissions (Aggiungi autorizzazioni).

Fase 1: Creazione di un bucket S3

Crea un bucket per archiviare i dati originali che intendi trasformare.

Per creare un bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Scegliere Create bucket (Crea bucket).

   Viene visualizzata la pagina Create bucket (Crea bucket).
4. In Bucket name (Nome bucket), inserisci un nome per il bucket (ad esempio **`tutorial-bucket`**).

   Per ulteriori informazioni sulle regole di denominazione del bucket in Amazon S3, consulta Regole di denominazione dei bucket.
5. In Region (Regione) scegli la Regione AWS in cui desideri che il bucket risieda.

   Per ulteriori informazioni sulla regione del bucket, consulta Panoramica dei bucket.
6. In Block Public Access settings for this bucket (Blocca le impostazioni di accesso pubblico per questo bucket), mantieni le impostazioni predefinite (è abilitato Block all public access (Blocca tutto l'accesso pubblico).

   È consigliabile di lasciare abilitate tutte le impostazioni di blocco dell'accesso pubblico, a meno che non abbia bisogno di disattivarne una o più per il caso d'uso. Per ulteriori informazioni sul blocco dell'accesso pubblico, consulta Blocco dell'accesso pubblico allo storage Amazon S3.
7. Mantieni le impostazioni rimanenti impostate sui valori di default.

   (Facoltativo) Se desideri configurare ulteriori impostazioni del bucket per il tuo caso d'uso specifico, consulta Creazione di un bucket.
8. Seleziona Crea bucket.

Fase 2: Caricamento di un file nel bucket S3

Carica un file di testo contenente dati PII noti di vari tipi, come nomi, informazioni bancarie, numeri di telefono e SSN, nel bucket S3 quali dati originali da cui oscurare i PII più avanti in questo tutorial.

Ad esempio, puoi caricare il seguente file `tutorial.txt`. Questo è un esempio di file di input di 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.
```

Per caricare un file in un bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckets (Bucket) scegli il nome del bucket creato nella Fase 1 (ad esempio,**`tutorial-bucket`**) in cui caricare il file.
4. Nella scheda Oggetti del bucket seleziona Carica.
5. Nella pagina Upload (Caricamento), sotto Files and Folders (File e cartelle) scegli Add Files (Aggiungi file).
6. Seleziona un file da caricare, quindi scegli Apri. Ad esempio, puoi caricare il file di esempio `tutorial.txt` menzionato in precedenza.
7. Scegli Carica.

Fase 3: Creazione di un punto di accesso S3

Per utilizzare un punto di accesso Lambda per oggetti S3 per accedere e trasformare i dati originali, devi creare un punto di accesso S3 e associarlo al bucket S3 creato nella Fase 1. Il punto di accesso deve trovarsi nello stesso punto in cui Regione AWS si trovano gli oggetti che desideri trasformare.

Più avanti in questo tutorial, utilizzerai questo punto di accesso come punto di accesso di supporto per il tuo punto di accesso Lambda per oggetti.

Per creare un punto di accesso

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Access Points (Punti di accesso).

3. Nella pagina Access Points (Punti di accesso) scegli Create access point (Crea punto di accesso).
4. Nel campo Access point name (Nome del punto di accesso), inserisci il nome (per esempio, **`tutorial-pii-access-point`**) per il punto di accesso.

   Per ulteriori informazioni sui punti di accesso S3, consulta Regole per la denominazione degli Punti di accesso Amazon S3.
5. Nel campo Bucket name (Nome bucket) inserisci il nome del bucket creato nella Fase 1 (ad esempio, **`tutorial-bucket`**). S3 allega quindi il punto di accesso a questo bucket.

   (Facoltativo) Puoi scegliere Browse S3 (Sfoglia S3) per sfogliare e cercare i bucket nell'account. Se scegli Browse S3 (Sfoglia S3), scegli il bucket desiderato e scegli Choose path (Scegli percorso) per popolare il campo Bucket name (Nome bucket) con il nome del bucket.
6. In Network origin (Origine rete), scegli Internet.

   Per ulteriori informazioni sulle origini di rete per i punti di accesso, consulta Creazione di access point limitati a un cloud privato virtuale.
7. Tutte le impostazioni di blocco dell'accesso pubblico sono abilitate per impostazione predefinita per il punto di accesso. È consigliabile lasciare abilitato Block all public access (Blocca tutto l'accesso pubblico). Per ulteriori informazioni, consulta Gestione dell'accesso pubblico agli access point.
8. Per tutte le altre impostazioni del punto di accesso, mantieni i valori di default.

   (Facoltativo) Puoi modificare le impostazioni del punto di accesso per supportare il caso d'uso. Per questo tutorial, ti consigliamo di mantenere le impostazioni di default.

   (Facoltativo) Se è necessario gestire l'accesso al punto di accesso, puoi specificare una policy per il punto di accesso. Per ulteriori informazioni, consulta Esempi di policy degli access point.
9. Selezionare Crea punto di accesso.

Fase 4: Configurazione e implementazione di una funzione Lambda precostituita

Per oscurare i dati PII, configura e implementa la funzione AWS Lambda precostituita `ComprehendPiiRedactionS3ObjectLambda` per l'utilizzo con il punto di accesso Lambda per oggetti S3.

Per configurare e implementare la funzione Lambda

1. Accedi a AWS Management Console e visualizza la `ComprehendPiiRedactionS3ObjectLambda`funzione in. AWS Serverless Application Repository
2. In Application settings (Impostazioni applicazioni), sotto Application name (Nome applicazione), mantieni il valore di default (`ComprehendPiiRedactionS3ObjectLambda`) per questo tutorial.

   (Facoltativo) Puoi inserire il nome che desideri assegnare a questa applicazione. Puoi eseguire questa operazione se prevedi di configurare più funzioni Lambda per esigenze di accesso diverse per lo stesso set di dati condiviso.
3. Per MaskCharacter, mantieni il valore predefinito (*****). Il carattere maschera sostituisce ogni carattere nell'entità PII oscurata.
4. Perché MaskMode, mantieni il valore predefinito (MASK). Il MaskModevalore specifica se l'entità PII viene oscurata con il `MASK` carattere o il valore. `PII_ENTITY_TYPE`
5. Per oscurare i tipi di dati specificati, for PiiEntityTypes, mantieni il valore predefinito ALL. Il PiiEntityTypesvalore specifica i tipi di entità PII da considerare per la redazione.

   Per ulteriori informazioni sull'elenco dei tipi di entità PII supportati, vedi Detect Personally Identifiable Information (PII) nella Guida per Developer di Amazon Comprehend.
6. Mantieni le impostazioni rimanenti impostate sui valori predefiniti.

   (Facoltativo) Se desideri configurare impostazioni aggiuntive per il caso d'uso specifico, consulta la sezione File readme sul lato sinistro della pagina.
7. Selezionare la casella di controllo accanto a I acknowledge that this app creates custom IAM roles (Confermo che questa app crea ruoli IAM personalizzati).
8. Seleziona Deploy (Implementa).
9. Nella pagina della nuova applicazione, sotto Resources (Risorse), scegli il Logical ID (ID logico) della funzione Lambda implementata per rivedere la funzione nella pagina della funzione Lambda.

Fase 5: Creazione di un punto di accesso Lambda per oggetti S3

Un punto di accesso Lambda per oggetti S3 offre la flessibilità di richiamare una funzione Lambda direttamente da una richiesta GET S3 in modo che la funzione possa oscurare i dati PII recuperati da un punto di accesso S3. Quando crei e configuri un punto di accesso Lambda per oggetti S3,

devi specificare la funzione Lambda di oscuramento da richiamare e fornire il contesto dell'evento in formato JSON come parametri personalizzati utilizzabili da Lambda.

Il contesto dell'evento fornisce informazioni relative alla richiesta eseguita nell'evento inviato da S3 Object Lambda a Lambda. Per ulteriori informazioni su tutti i campi nel contesto dell'evento, consulta Formato e utilizzo del contesto degli eventi.

Per creare un punto di accesso Lambda per oggetti S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Object Lambda Access Points (Punti di accesso Object Lambda).
3. Nella pagina Object Lambda Access Points (Punti di accesso Object Lambda), scegli Create Object Lambda Access Point (Crea punto di accesso Object Lambda).
4. In Nome del punto di accesso per le espressioni Lambda dell'oggetto immetti il nome che desideri utilizzare per il punto di accesso Lambda per oggetti (per esempio, **`tutorial-pii-object-lambda-accesspoint`**).
5. In Supporting Access Point (Access point di supporto), inserisci o seleziona il punto di accesso standard creato nella Fase 3 (ad esempio, **`tutorial-pii-access-point`**), quindi scegli Choose supporting Access Point (Scegli punto di accesso di supporto).
6. Per le API S3, per recuperare gli oggetti dal bucket S3 per l'elaborazione della funzione Lambda, seleziona. GetObject
7. In Invoke Lambda function (Chiama una funzione Lambda) per questo tutorial puoi scegliere una delle due opzioni seguenti.
   - Scegli Choose from functions in your account (Scegli tra le funzioni nell'account) e scegli la funzione Lambda implementata nella Fase 4 (ad esempio, **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**) dall'elenco a discesa Lambda function (Funzione Lambda).
   - Scegli Enter ARN (Inserisci ARN), quindi inserisci l'Amazon Resource Name (ARN) della funzione Lambda creata nella Fase 4.
8. In Lambda function version (Versione delle funzioni Lambda), scegli $LATEST (l'ultima versione della funzione Lambda implementata nella Fase 4).
9. (Facoltativo) Se hai bisogno che la tua funzione Lambda riconosca ed elabori le richieste GET con intestazioni con intervalli e numeri di parte, seleziona Lambda function supports requests

using range (La funzione Lambda supporta le richieste che utilizzano l'intervallo) e Lambda function supports requests using part numbers (La funzione Lambda supporta le richieste che utilizzano numeri di parte). Altrimenti, deselezionare queste due caselle di controllo.

Per ulteriori informazioni sull'utilizzo di intervalli o numeri di parte con S3 Object Lambda, consulta Utilizzo delle intestazioni Range e partNumber.

10. (Facoltativo) In Payload - optional (Payload - facoltativo), aggiungi il testo JSON per fornire alla tua funzione Lambda ulteriori informazioni.

    Un payload è un testo JSON opzionale che puoi fornire alla tua funzione Lambda come input per tutte le chiamate provenienti da uno specifico punto di accesso Lambda per oggetti S3. Per personalizzare il comportamento di più punti di accesso Lambda per oggetti che richiamano la stessa funzione Lambda, puoi configurare i payload con parametri diversi, estendendo così la flessibilità della funzione stessa.

    Per ulteriori informazioni sul payload, consulta Formato e utilizzo del contesto degli eventi.

11. (Facoltativo) In Parametri di richiesta - facoltativo, scegli Disabilita o Abilita per aggiungere il monitoraggio Amazon S3 al punto di accesso Lambda per oggetti. Le metriche delle richieste vengono fatturate alla tariffa standard di Amazon CloudWatch . Per ulteriori informazioni, consultare Prezzi di CloudWatch.
12. In Object Lambda Access Point policy - optional (Policy del punto di accesso Object Lambda - facoltativo mantieni l'impostazione di default.

    (Facoltativo) Puoi impostare una policy delle risorse. Questa policy delle risorse fornisce all'API `GetObject` l'autorizzazione per utilizzare il punto di accesso Lambda per oggetti specificato.

13. Mantieni le impostazioni rimanenti sui valori di default, quindi scegliCreate Object Lambda Access Point (Crea punto di accesso Object Lambda).

Fase 6: Utilizzo del punto di accesso Lambda per oggetti S3 per recuperare il file oscurato

Ora, S3 Object Lambda è pronto a oscurare i dati PII dal file originale.

Per utilizzare il punto di accesso Lambda per oggetti S3 per recuperare il file oscurato

Quando chiedi di recuperare un file tramite il punto di accesso Lambda per oggetti S3, esegui una chiamata API `GetObject` a Lambda per oggetti S3. S3 Object Lambda richiama la funzione Lambda

per oscurare i dati PII e restituisce i dati trasformati come risposta alla chiamata API `GetObject` S3 standard.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Object Lambda Access Points (Punti di accesso Object Lambda).
3. Nella pagina Punti di accesso Lambda dell'oggetto scegli il punto di accesso Lambda per oggetti S3 creato nella Fase 5 (ad esempio, **`tutorial-pii-object-lambda-accesspoint`**).
4. Nella scheda Oggetti del punto di accesso Lambda per oggetti S3, seleziona il file con lo stesso nome (ad esempio, `tutorial.txt`) di quello che hai caricato nel bucket S3 nella Fase 2.

   Questo file deve contenere tutti i dati trasformati.
5. Per visualizzare i dati trasformati, scegli Open (Apri) o Download (Scarica).

   Dovresti visualizzare il file oscurato, come mostrato nell'esempio seguente.

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

Fase 7: pulire

Se hai oscurato i tuoi dati tramite S3 Object Lambda solo come esercizio di apprendimento, elimina le AWS risorse che hai allocato in modo da non incorrere più in costi.

Fasi secondarie

- Eliminazione del punto di accesso Lambda per oggetti

- Eliminazione del punto di accesso S3
- Eliminazione della funzione Lambda
- Eliminare il gruppo di CloudWatch log
- Eliminazione del file originale nel bucket S3 di origine
- Eliminazione del bucket S3 di origine
- Eliminazione del ruolo IAM per la funzione Lambda
- Eliminazione dei criteri gestiti dal cliente per l'utente IAM
- Eliminazione dell'utente IAM

Eliminazione del punto di accesso Lambda per oggetti

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Object Lambda Access Points (Punti di accesso Object Lambda).
3. Nella pagina Punti di accesso Lambda dell'oggetto scegli il pulsante di opzione a sinistra del punto di accesso Lambda per oggetti S3 creato nella Fase 5 (ad esempio, **`tutorial-pii-object-lambda-accesspoint`**).
4. Scegli Elimina.
5. Conferma di voler eliminare il punto di accesso Lambda per oggetti inserendone il nome nel campo di testo che viene visualizzato, quindi scegli Elimina.

Eliminazione del punto di accesso S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Access Points (Punti di accesso).
3. Passa al punto di accesso creato nella Fase 3 (ad esempio, **`tutorial-pii-access-point`**), quindi scegli il pulsante di opzione accanto al nome del punto di accesso.
4. Scegli Elimina.
5. Conferma di voler eliminare il punto di accesso inserendone il nome nel campo di testo che viene visualizzato, quindi scegli Delete (Elimina).

Eliminazione della funzione Lambda

1. Nella AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/, scegli Funzioni nel riquadro di navigazione a sinistra.
2. Scegli la funzione creata nella Fase 4 (ad esempio, **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**).
3. Scegli Azioni, quindi Elimina.
4. Nella finestra di dialogo Delete function (Elimina funzione), scegli Delete (Elimina).

Eliminare il gruppo di CloudWatch log

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione a sinistra, scegli Log groups (Gruppi di registri).
3. Individua il gruppo di registri il cui nome termina con la funzione Lambda creata nella Fase 4(ad esempio, **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**).
4. Scegli Actions (Operazioni), quindi scegli Delete log group(s) (Elimina gruppi di registri).
5. Nella finestra di dialogo Delete log group(s) (Elimina gruppo/i di log) scegli Delete (Elimina).

Eliminazione del file originale nel bucket S3 di origine

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket name (Nome bucket) scegli il nome del bucket su cui hai caricato il file originale nella Fase 2 (ad esempio, **`tutorial-bucket`**).
4. Seleziona la casella di controllo a sinistra del nome dell'oggetto da eliminare (ad esempio, `tutorial.txt`).
5. Scegli Elimina.
6. Nella pagina Delete objects (Elimina oggetti), nella sezione Permanently delete objects? (Eliminare definitivamente gli oggetti?) conferma che desideri eliminare questo oggetto inserendo **`permanently delete`** nella casella di testo.
7. Scegliere Delete objects (Elimina oggetti).

Eliminazione del bucket S3 di origine

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckets (Bucket) scegli il pulsante di opzione accanto al nome del bucket creato nella Fase 1(ad esempio,**`tutorial-bucket`**).
4. Scegli Elimina.
5. Nella pagina Delete bucket (Elimina bucket) conferma che desideri eliminare il bucket inserendone il nome nel campo di testo e quindi scegli Delete bucket (Elimina bucket).

Eliminazione del ruolo IAM per la funzione Lambda

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, scegli Roles (Ruoli), quindi seleziona la casella di controllo accanto al nome del ruolo che desideri eliminare. Il nome del ruolo inizia con il nome della funzione Lambda implementata nella Fase 4 (ad esempio, **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**).
3. Scegli Elimina.
4. Nella casella di dialogo Delete (Elimina), inserisci il nome del ruolo nel campo di inserimento del testo per confermare l'eliminazione. Quindi, scegli Elimina.

Eliminazione dei criteri gestiti dal cliente per l'utente IAM

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra, seleziona Policy.
3. Nella pagina Policies (Policy) inserisci il nome della policy gestita dal cliente creata nei Prerequisiti (ad esempio, **`tutorial-serverless-application-repository`**) nella casella di ricerca per filtrare l'elenco di policy. Seleziona il pulsante di opzione accanto al nome della policy che desideri eliminare.
4. Scegli Azioni, quindi Elimina.

5. Conferma di voler eliminare questa policy inserendone il nome nel campo di testo che viene visualizzato, quindi scegli Delete (Elimina).

Eliminazione dell'utente IAM

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra, scegli Users (Utenti), quindi selezionare la casella di controllo accanto al nome utente che desideri eliminare.
3. Nella parte superiore della pagina, scegli Delete (Elimina).
4. Nella casella di dialogo Delete ***user name***? (Eliminare nome utente?) inserisci il nome utente nel campo di inserimento del testo per confermare l'eliminazione dell'utente. Scegli Elimina.

Passaggi successivi

Dopo aver completato questo tutorial, puoi esplorare altri casi d'uso correlati:

- Puoi creare più punti di accesso Lambda per oggetti S3 e abilitarli con funzioni Lambda precostituite configurate in modo diverso per oscurare specifici tipi di PII a seconda delle esigenze aziendali di chi accede ai dati.

  Ogni tipo di utente assume un ruolo IAM e ha accesso solo a un punto di accesso Lambda per oggetti S3 (gestito tramite policy IAM). Quindi, collega ogni funzione Lambda `ComprehendPiiRedactionS3ObjectLambda` configurata per un diverso caso d'uso di oscuramento a un diverso punto di accesso Lambda per oggetti S3. Per ogni punto di accesso Lambda per oggetti S3, puoi disporre di un punto di accesso S3 di supporto per leggere i dati da un bucket S3 che archivia il set di dati condiviso.

  Per ulteriori informazioni su come creare una policy di bucket S3 che consenta agli utenti di leggere dal bucket solo tramite i punti di accesso S3, consulta Configurazione delle policy IAM per l'utilizzo degli access point.

  Per ulteriori informazioni su come concedere a un utente l'autorizzazione per accedere alla funzione Lambda, al punto di accesso S3 e al punto di accesso Lambda per oggetti S3, consulta Configurazione delle policy IAM per i punti di accesso Lambda per oggetti.
- Puoi creare una funzione Lambda personalizzata e utilizzarla in S3 Object Lambda per soddisfare le tue esigenze specifiche relative ai dati.

Ad esempio, per esplorare vari valori di dati, puoi utilizzare S3 Object Lambda e la funzione Lambda personalizzata che utilizza ulteriori funzionalità di Amazon Comprehend, come il riconoscimento delle entità, il riconoscimento delle frasi chiave, l'analisi del sentimento e la classificazione dei documenti, per elaborare i dati. Puoi utilizzare S3 Object Lambda insieme a Amazon Comprehend Medical, un servizio di NLP idoneo per HIPAA, per analizzare ed estrarre i dati in modo contestuale.

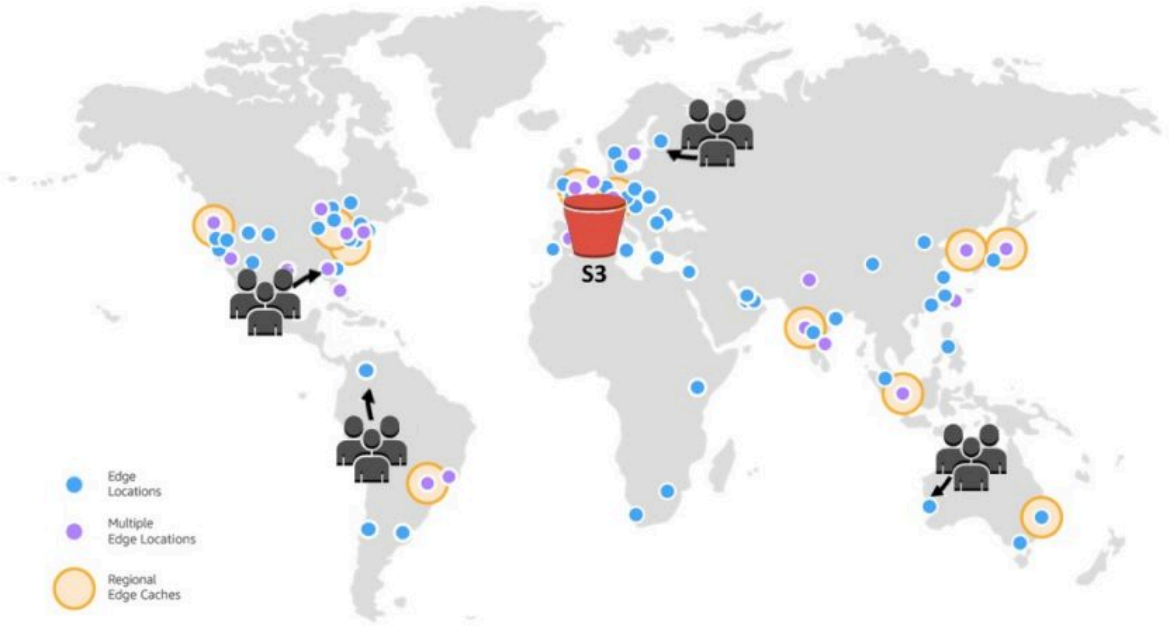
Per ulteriori informazioni su come trasformare i dati con S3 Object Lambda e sulla funzione Lambda personalizzata, consulta Tutorial: trasformazione dei dati per l'applicazione con S3 Object Lambda.

Tutorial: hosting di video in streaming su richiesta con Amazon S3, Amazon e CloudFront Amazon Route 53

Puoi usare Amazon S3 con Amazon CloudFront per ospitare video per la visualizzazione su richiesta in modo sicuro e scalabile. Nello streaming di video on demand (VOD), i contenuti video vengono archiviati su un server e gli spettatori possono guardarli in qualsiasi momento.

CloudFront è un servizio di rete per la distribuzione di contenuti (CDN) veloce, altamente sicuro e programmabile. CloudFront può distribuire i tuoi contenuti in modo sicuro tramite HTTPS da tutte le CloudFront edge location in tutto il mondo. Per ulteriori informazioni su CloudFront, consulta What is Amazon CloudFront? nella Amazon CloudFront Developer Guide.

CloudFront la memorizzazione nella cache riduce il numero di richieste a cui il server di origine deve rispondere direttamente. Quando uno spettatore (utente finale) richiede un video con cui servite CloudFront, la richiesta viene indirizzata a una location periferica più vicina a dove si trova lo spettatore. CloudFront serve il video dalla sua cache, recuperandolo dal bucket S3 solo se non è già memorizzato nella cache. Ciò accelera la distribuzione dei video agli spettatori a livello globale con bassa latenza e velocità effettiva e velocità di trasferimento elevate. Per ulteriori informazioni sulla gestione della CloudFront cache, consulta Optimizing caching and availability nella Amazon CloudFront Developer Guide.



Obiettivo

In questo tutorial, configurerai un bucket S3 per ospitare lo streaming video su richiesta utilizzando CloudFront for delivery e Amazon Route 53 per Domain Name System (DNS) e la gestione personalizzata del dominio.

Argomenti

Prerequisiti: registrazione e configurazione di un dominio personalizzato con Route 53

Prima di iniziare questo tutorial, devi registrare e configurare un dominio personalizzato (ad esempio,**`example.com`**) con Route 53 in modo da poter configurare la CloudFront distribuzione per utilizzare un nome di dominio personalizzato in un secondo momento.

Senza un nome di dominio personalizzato, il tuo video S3 è accessibile pubblicamente e ospitato tramite CloudFront un URL simile al seguente:

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

Ad esempio, **`https://d111111abcdef8.cloudfront.net/sample.mp4`**.

Dopo aver configurato la CloudFront distribuzione per utilizzare un nome di dominio personalizzato configurato con Route 53, il video S3 è accessibile pubblicamente e ospitato tramite CloudFront un URL simile al seguente:

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

Ad esempio, **`https://www.example.com/sample.mp4`**. Un nome di dominio personalizzato è più semplice e intuitivo da usare per gli spettatori.

Per registrare un nome di dominio, consulta Registrazione dei nomi di dominio utilizzando Route 53 nella Guida per gli sviluppatori di Amazon Route 53.

Quando registri un nome di dominio con Route 53, Route 53 crea automaticamente la zona ospitata, che utilizzerai più avanti in questo tutorial. Questa zona ospitata è il luogo in cui memorizzi le informazioni su come indirizzare il traffico per il tuo dominio, ad esempio, verso un'istanza o una CloudFront distribuzione Amazon EC2.

Sono previste tariffe associate alla registrazione del dominio, alla tua zona ospitata e alle query DNS ricevute dal tuo dominio. Per ulteriori informazioni, consulta la pagina dei Prezzi Amazon Route 53.

> **Note**
>
> Quando registri un dominio, il costo è immediato ed è irreversibile. Puoi scegliere di non rinnovare automaticamente il dominio, ma il pagamento è anticipato e resti proprietario per un anno. Per maggiori informazioni, consulta Registrazione di un nuovo dominio nella Guida per gli sviluppatori di Amazon Route 53.

Fase 1: Creazione di un bucket S3

Devi creare un bucket per archiviare il video originale che intendi riprodurre in streaming.

Per creare un bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Scegliere Create bucket (Crea bucket).

   Viene visualizzata la pagina Create bucket (Crea bucket).
4. Per Nome bucket, immetti un nome per il bucket, ad esempio **`tutorial-bucket`**.

   Per ulteriori informazioni sulle regole di denominazione del bucket in Amazon S3, consulta Regole di denominazione dei bucket.
5. Per Regione, scegli Regione AWS dove vuoi che risieda il bucket.

   Se possibile, dovresti scegliere la località della regione che probabilmente sarà più vicina alla maggior parte dei tuoi spettatori. Per ulteriori informazioni sulla regione del bucket, consulta Panoramica dei bucket.
6. In Block Public Access settings for this bucket (Blocca le impostazioni di accesso pubblico per questo bucket), mantieni le impostazioni predefinite (è abilitato Block all public access (Blocca tutto l'accesso pubblico).

   Anche se l'opzione Blocca tutti gli accessi pubblici è abilitata, gli spettatori possono comunque accedere al video caricato tramite. CloudFront Questa funzionalità è uno dei principali vantaggi dell'utilizzo CloudFront per ospitare un video archiviato in S3.

È consigliabile di lasciare tutte le impostazioni abilitate, a meno che non abbia bisogno di disattivarne una o più per il caso d'uso. Per ulteriori informazioni sul blocco dell'accesso pubblico, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

7. Mantieni le impostazioni rimanenti impostate sui valori di default.

   (Facoltativo) Se desideri configurare ulteriori impostazioni del bucket per il tuo caso d'uso specifico, consulta Creazione di un bucket.

8. Seleziona Crea bucket.

Fase 2: Caricamento di un video nel bucket S3

La procedura riportata di seguito illustra come caricare un file video in un bucket S3 utilizzando la console. Quando carichi un video in S3, puoi anche possibile utilizzare Amazon S3 Transfer Acceleration per configurare trasferimenti di file veloci e sicuri. Transfer Acceleration può velocizzare il caricamento dei video nel bucket S3 per il trasferimento a lunga distanza di video di grandi dimensioni. Per ulteriori informazioni, consulta Configurazione di trasferimenti veloci e sicuri di file con Amazon S3 Transfer Acceleration.

Per caricare un file nel bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckets (Bucket) scegli il nome del bucket creato nella Fase 1 (ad esempio,**`tutorial-bucket`**) in cui caricare il file.
4. Nella scheda Oggetti del bucket seleziona Carica.
5. Nella pagina Upload (Caricamento), sotto Files and Folders (File e cartelle) scegli Add Files (Aggiungi file).
6. Seleziona un file da caricare, quindi scegli Apri.

   Ad esempio, puoi caricare un file video denominato `sample.mp4`.

7. Scegli Carica.

Fase 3: Creare un'identità di accesso all' CloudFront origine

Per limitare l'accesso diretto al video dal tuo bucket S3, crea un CloudFront utente speciale chiamato Origin Access Identity (OAI). In questo tutorial, assocerai l'OAI alla distribuzione. Utilizzando un OAI, ti assicuri che gli spettatori non possano ignorarlo CloudFront e ricevere il video direttamente dal bucket S3. Solo l' CloudFront OAI può accedere al file nel bucket S3. Per ulteriori informazioni, consulta Limitazione dell'accesso ai contenuti di Amazon S3 utilizzando un OAI nella Amazon CloudFront Developer Guide.

Per creare un OAI CloudFront

1. Accedi a AWS Management Console e apri la CloudFront console all'indirizzohttps://console.aws.amazon.com/cloudfront/v4/home.
2. Nel pannello di navigazione sulla sinistra, nella sezione Sicurezza, scegli Accesso origine.
3. Nella scheda Identità, scegli Crea identità di accesso origine.
4. ImmInserisci un nome (ad esempio, **`S3-OAI`**) come nuova identità di accesso origine.
5. Scegli Crea.

Fase 4: Creare una CloudFront distribuzione

Per CloudFront utilizzarlo per servire e distribuire il video nel tuo bucket S3, devi creare una CloudFront distribuzione.

Fasi secondarie

- Crea una distribuzione CloudFront
- Revisione della policy del bucket

Crea una distribuzione CloudFront

1. Accedi a AWS Management Console e apri la CloudFront console all'indirizzohttps://console.aws.amazon.com/cloudfront/v4/home.
2. Nel riquadro di navigazione a sinistra, scegliereDistribuzioni.
3. Scegli Create Distribution (Crea distribuzione).
4. Nella sezione Origine, per Dominio origine scegli il nome di dominio dell'origine S3, che inizia con il nome del bucket S3 creato nella Fase 1 (ad esempio, **`tutorial-bucket`**).

5. Per Accesso origine, seleziona Identità di accesso legacy.
6. In Identità di accesso origine, scegli l'identità di accesso all'origine esistente creata nella Fase 3 (ad esempio, **S3-OAI**).
7. In Bucket policy (Policy del bucket), scegli Yes, update the bucket policy (Sì, aggiorna la policy del bucket).
8. In Funzionamento cache predefinito, nella sezione Policy protocollo visualizzatore, scegli Reindirizza HTTP a HTTPS.

   Questo significa che le richieste HTTP vengono reindirizzate automaticamente a HTTPS per proteggere il tuo sito Web e proteggere i dati degli spettatori.
9. Per le altre impostazioni nella sezione Default Cache Behavior Settings (Modifica impostazioni comportamento cache), accettare i valori predefiniti.

   (Facoltativo) Puoi controllare per quanto tempo il file rimane nella CloudFront cache prima di CloudFront inoltrare un'altra richiesta all'origine. Riducendo la durata, puoi distribuire contenuti dinamici. Aumentando la durata, i visualizzatori otterranno prestazioni migliori, poiché è più probabile che i file vengano distribuiti direttamente dalla cache edge. Una durata maggiore riduce anche il carico sul server di origine. Per ulteriori informazioni, consulta Gestione della durata della permanenza dei contenuti nella cache (scadenza) nella Amazon CloudFront Developer Guide.
10. Per le altre sezioni, mantieni le impostazioni rimanenti impostate sui valori predefiniti.

    Per ulteriori informazioni sulle diverse opzioni di impostazione, consulta Valori che specifichi quando crei o aggiorni una distribuzione nella Amazon CloudFront Developer Guide.
11. Nella parte inferiore della pagina, scegli Create distribution (Crea distribuzione).
12. Nella scheda Generale della tua CloudFront distribuzione, in Dettagli, il valore della colonna Ultima modifica per la tua distribuzione cambia da Distribuzione al timestamp dell'ultima modifica della distribuzione. In genere sono necessari pochi minuti.

Revisione della policy del bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).

3. Nell'elenco dei bucket, scegli il nome del bucket che hai usato in precedenza come origine della tua CloudFront distribuzione (ad esempio,). **`tutorial-bucket`**
4. Scegli la scheda Autorizzazioni.
5. Nella casella di testo Bucket policy (Policy del bucket) conferma di visualizzare una formulazione simile alla seguente:

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

Questa è l'affermazione che la tua CloudFront distribuzione ha aggiunto alla tua policy sui bucket quando hai scelto Sì, aggiorna prima la policy del bucket.

Questo aggiornamento della policy sui bucket indica che hai configurato correttamente la CloudFront distribuzione per limitare l'accesso al bucket S3. A causa di questa restrizione, è possibile accedere agli oggetti nel bucket solo tramite la tua distribuzione. CloudFront

Passaggio 5: Accedi al video tramite la distribuzione CloudFront

Ora CloudFront puoi servire il video memorizzato nel tuo bucket S3. Per accedere al video tramite CloudFront, devi combinare il nome del dominio di CloudFront distribuzione con il percorso del video nel bucket S3.

Per creare un URL per il video S3 utilizzando il nome del CloudFront dominio di distribuzione

1. Accedi a AWS Management Console e apri la CloudFront console all'indirizzohttps://console.aws.amazon.com/cloudfront/v4/home.
2. Nel riquadro di navigazione a sinistra, scegliereDistribuzioni.
3. Per ottenere il nome di dominio della distribuzione, procedi come indicato di seguito:

   a. Nella colonna Origins, trova la CloudFront distribuzione corretta cercando il nome di origine, che inizia con il bucket S3 che hai creato nel passaggio 1 (ad esempio,**`tutorial-bucket`**).

   b. Dopo aver trovato la distribuzione nell'elenco, amplia la colonna Nome di dominio per copiare il valore del nome di dominio per la tua distribuzione. CloudFront
4. In una nuova scheda del browser, incolla il nome di dominio della distribuzione copiato in precedenza.
5. Torna alla scheda precedente del browser, quindi apri la console S3 all'indirizzo https://console.aws.amazon.com/s3/.
6. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
7. Nell'elenco Buckets (Bucket) scegli il nome del bucket creato nella Fase 1 (ad esempio, **`tutorial-bucket`**).
8. Nell'elenco Objects (Oggetti) scegli il nome del video che hai caricato nella Fase 2 ai fini dello streaming (ad esempio, `sample.mp4`).
9. Nella pagina prodotto dell'oggetto, nellaPanoramica dell'oggettosezione, copiare il valore della Chiave. Questo valore è il percorso dell'oggetto video caricato nel bucket S3.
10. Torna alla scheda del browser in cui hai precedentemente incollato il nome del dominio di distribuzione, inserisci una barra di inoltro (**`/`**) dopo il nome del dominio di distribuzione, quindi incolla il percorso al video copiato in precedenza (ad esempio,`sample.mp4`).

    Ora, il tuo video S3 è accessibile pubblicamente e ospitato tramite CloudFront un URL simile al seguente:

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

    Sostituisci *`il nome del dominio di CloudFront distribuzione`* e *`il percorso del video S3`* con i valori appropriati. Un esempio di URL è **`https://d111111abcdef8.cloudfront.net/sample.mp4`**.

Passaggio 6: configura la CloudFront distribuzione per utilizzare il nome di dominio personalizzato

Per utilizzare il tuo nome di dominio anziché il nome di CloudFront dominio nell'URL per accedere al video S3, aggiungi un nome di dominio alternativo alla tua CloudFront distribuzione.

Fasi secondarie

- Richiesta di un certificato SSL
- Aggiungi il nome di dominio alternativo alla tua distribuzione CloudFront
- Crea un record DNS per indirizzare il traffico dal tuo nome di dominio alternativo al nome di dominio della tua distribuzione CloudFront
- Controllo dell'abilitazione di IPv6 per la distribuzione ed eventuale creazione di un altro registro DNS

Richiesta di un certificato SSL

Per consentire ai tuoi spettatori di utilizzare HTTPS e il tuo nome di dominio personalizzato nell'URL per lo streaming video, utilizza AWS Certificate Manager (ACM) per richiedere un certificato Secure Sockets Layer (SSL). Il certificato SSL stabilisce una connessione di rete crittografata al sito Web.

1. Accedi AWS Management Console e apri la console ACM all'indirizzo https://console.aws.amazon.com/acm/.
2. Se viene visualizzata la pagina introduttiva, in Provision certificates (Fornisci certificati), scegli Get Started (Inizia).
3. Nella pagina Richiedi un certificato scegli Richiedi un certificato pubblico e poi di nuovo Richiedi un certificato.
4. Nella pagina Add domain names (Aggiungi nomi di dominio) digita il nome di dominio completo del sito che desideri proteggere con un certificato SSL/TLS. Utilizza un asterisco (*) per richiedere un certificato jolly che protegge diversi nomi di siti nello stesso dominio. In questo tutorial, digita ***** e il nome di dominio personalizzato configurato in Prerequisiti. Ad esempio, immettere***.example.com**e quindi sceglisuccessivo.

   Per ulteriori informazioni, consulta Per richiedere un certificato pubblico ACM (console) nella Guida per l'utente di AWS Certificate Manager .
5. Nella pagina Select validation method (Seleziona metodo di convalida), scegli DNS validation (Convalida DNS). Quindi, seleziona Next (Successivo).

Se si è in grado di modificare la configurazione DNS, si consiglia di utilizzare la convalida del dominio DNS anziché la convalida email. La convalida del DNS offre diversi vantaggi rispetto alla convalida dell'email. Per ulteriori informazioni, consultaOpzione 1: convalida DNSnellaAWS Certificate Manager Guida per l'utente di.

6. (Facoltativo) Nella pagina Aggiungi tag puoi contrassegnare facoltativamente il certificato con metadati.
7. Scegli Rivedi.
8. Nella pagina Revisione, verifica che le informazioni presenti in Nome dominio e Metodo di convalida siano corrette. Dopodiché, seleziona Confirm and request (Conferma e richiedi).

   La pagina Convalida mostra che la richiesta è in fase di elaborazione e che il dominio certificato viene convalidato. I certificati in attesa di convalida hanno lo stato Pending validation (Convalida in attesa).
9. Nella pagina Convalida, scegli la freccia verso il basso a sinistra del nome di dominio personalizzato e seleziona Crea registro in Route 53 per convalidare la proprietà del dominio tramite DNS.

   In questo modo viene aggiunto un record CNAME fornito da AWS Certificate Manager alla configurazione DNS.
10. Nella casella di dialogo Create record in Route 53 (Crea registro in Route 53), scegli Create (Crea).

    La pagina Convalida dovrebbe ora visualizzare la notifica di stato Riuscito in basso.
11. Scegli Continue (Continua) per visualizzare la pagina elenco Certificates (Certificati).

    Lo Stato del nuovo certificato passerà da Convalida in attesa a Emesso entro 30 minuti.

Aggiungi il nome di dominio alternativo alla tua distribuzione CloudFront

1. Accedi a AWS Management Console e apri la CloudFront console all'indirizzohttps://console.aws.amazon.com/cloudfront/v4/home.
2. Nel riquadro di navigazione a sinistra, scegliereDistribuzioni.
3. Scegli l'ID della distribuzione creata nella Fase 4.
4. SulGeneralitab, vai alImpostazionisezione, e scegliModificare.

5. Nella pagina Modifica impostazioni, per Nome di dominio alternativo (CNAME), facoltativo, scegli Aggiungi elemento per aggiungere i nomi di dominio personalizzati che desideri utilizzare nell'URL del video S3 servito da questa distribuzione. CloudFront

   In questo tutorial, ad esempio, se desideri instradare il traffico a un sottodominio, ad esempio `www.example.com`, inserisci il nome del sottodominio (`www`) con il nome di dominio (`example.com`). In particolare, inserisci **`www.example.com`**.

   > **Note**
   >
   > Il nome di dominio alternativo (CNAME) che aggiungi deve essere coperto dal certificato SSL che hai precedentemente allegato alla tua distribuzione. CloudFront

6. In Certificato SSL personalizzato - facoltativo, scegli il certificato SSL richiesto in precedenza (ad esempio, **`*.example.com`**).

   > **Note**
   >
   > Se il certificato SSL non viene visualizzato immediatamente dopo averlo richiesto, attendi 30 minuti, quindi aggiorna l'elenco fino a quando il certificato SSL diventa disponibile per la selezione.

7. Mantieni le impostazioni rimanenti impostate sui valori predefiniti. Seleziona Salvataggio delle modifiche.
8. Nella scheda Generale per la distribuzione, attendi che il valore di Ultima modifica passi da Implementazione in corso al timestamp dell'ultima modifica della distribuzione.

Crea un record DNS per indirizzare il traffico dal tuo nome di dominio alternativo al nome di dominio della tua distribuzione CloudFront

1. Accedi AWS Management Console e apri la console Route 53 all'indirizzo https://console.aws.amazon.com/route53/.
2. Nel riquadro di navigazione a sinistra, scegliere Hosted zones (Zone ospitate).
3. Nella pagina Zone ospitate seleziona il nome della zona ospitata creata da Route 53 in Prerequisiti (ad esempio, **`example.com`**).
4. ScegliereCreare recorde quindi usa ilCreazione rapida recordmetodo.

5. Per Record name, mantieni il valore del nome del record uguale al nome di dominio alternativo della CloudFront distribuzione che hai aggiunto in precedenza.

   In questo tutorial, per instradare il traffico a un sottodominio, ad esempio `www.example.com`, inserisci il nome del sottodominio senza il nome di dominio. Ad esempio, inserisci solo **`www`** nel campo di testo prima del nome di dominio personalizzato.
6. Per Tipo di record, scegli A - Indirizza il traffico verso un indirizzo IPv4 e alcune risorse. AWS
7. In Valore, scegli l'attivazione/disattivazione Alias per abilitare la risorsa Alias.
8. In Indirizza il traffico verso, scegli Alias per la CloudFront distribuzione dall'elenco a discesa.
9. Nella casella di ricerca che dice Scegli la distribuzione, scegli il nome di dominio della CloudFront distribuzione che hai creato nel passaggio 4.

   Per trovare il nome di dominio della tua CloudFront distribuzione, procedi come segue:

   a. In una nuova scheda del browser, accedi AWS Management Console e apri la CloudFront console all'indirizzohttps://console.aws.amazon.com/cloudfront/v3/home.
   b. Nel riquadro di navigazione a sinistra, scegliereDistribuzioni.
   c. Nella colonna Origins, trova la CloudFront distribuzione corretta cercando il nome di origine, che inizia con il bucket S3 che hai creato nel passaggio 1 (ad esempio,**`tutorial-bucket`**).
   d. Dopo aver trovato la distribuzione nell'elenco, amplia la colonna Nome di dominio per visualizzare il valore del nome di dominio per la tua distribuzione. CloudFront
10. SulCreare recordNella console Route 53, per le impostazioni rimanenti, mantenere i valori predefiniti.
11. Scegli Crea record.

Controllo dell'abilitazione di IPv6 per la distribuzione ed eventuale creazione di un altro registro DNS

Se IPv6 è abilitato per la distribuzione, devi creare un altro registro DNS.

1. Per verificare se IPv6 è abilitato per la distribuzione, procedi come indicato di seguito:

   a. Accedi a AWS Management Console e apri la CloudFront console all'indirizzohttps://console.aws.amazon.com/cloudfront/v4/home.
   b. Nel riquadro di navigazione a sinistra, scegliereDistribuzioni.
   c. Scegli l'ID della CloudFront distribuzione che hai creato nel passaggio 4.

d. Nella scheda Generale, in Impostazioni, controlla se IPv6 è impostato su Abilitato.

   Se IPv6 è abilitato per la distribuzione, devi creare un altro registro DNS.

2. Se IPv6 è abilitato per la distribuzione, crea un registro DNS eseguendo le seguenti operazioni:

   a. Accedi AWS Management Console e apri la console Route 53 all'indirizzo https://console.aws.amazon.com/route53/.

   b. Nel riquadro di navigazione a sinistra, scegliere Hosted zones (Zone ospitate).

   c. Nella pagina Zone ospitate seleziona il nome della zona ospitata creata da Route 53 in Prerequisiti (ad esempio, **`example.com`**).

   d. ScegliereCreare recorde quindi usa ilCreazione rapida recordmetodo.

   e. In Nome registro, nel campo di testo che precede il nome di dominio personalizzato, digita lo stesso valore inserito quando hai creato in precedenza il registro DNS IPv4. Ad esempio, in questo tutorial, per instradare il traffico a `www.example.com`, inserisci solo **`www`**.

   f. In Record type (Tipo di registro), scegli AAAA - Routes traffic to an IPv6 address and some AWS resources (AAAA - Instrada il traffico su un indirizzo IPv6 e su alcune risorse AWS).

   g. In Valore, scegli l'attivazione/disattivazione Alias per abilitare la risorsa Alias.

   h. In Route traffic to, scegli Alias to CloudFront distribution dall'elenco a discesa.

   i. Nella casella di ricerca che dice Scegli la distribuzione, scegli il nome di dominio della CloudFront distribuzione che hai creato nel passaggio 4.

   j. Mantieni le impostazioni rimanenti impostate sui valori di default.

   k. Scegli Crea record.

Passaggio 7: accedi al video S3 tramite la CloudFront distribuzione con il nome di dominio personalizzato

Per accedere al video S3 utilizzando l'URL personalizzato, devi combinare il nome di dominio alternativo con il percorso del video nel bucket S3.

Per creare un URL personalizzato per accedere al video S3 tramite la distribuzione CloudFront

1. Accedi a AWS Management Console e apri la CloudFront console all'indirizzohttps://console.aws.amazon.com/cloudfront/v4/home.

2. Nel riquadro di navigazione a sinistra, sceglioreDistribuzioni.

3. Per ottenere il nome di dominio alternativo della tua CloudFront distribuzione, procedi come segue:

   a. Nella colonna Origins, trova la CloudFront distribuzione corretta cercando il nome di origine, che inizia con il nome del bucket S3 per il bucket che hai creato nel passaggio 1 (ad esempio,). **`tutorial-bucket`**

   b. Dopo aver trovato la distribuzione nell'elenco, amplia la colonna Nomi di dominio alternativi per copiare il valore del nome di dominio alternativo della tua distribuzione. CloudFront

4. In una nuova scheda del browser, incolla il nome di dominio alternativo della distribuzione. CloudFront

5. Torna alla scheda precedente del browser, quindi apri la console di Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

6. Trova il percorso per il video S3, come spiegato nella Fase 5.

7. Torna alla scheda del browser in cui hai precedentemente incollato il nome di dominio alternativo, continua digitando **`/`** e incolla il percorso al video S3 (ad esempio, `sample.mp4`).

   Ora, il tuo video S3 è accessibile CloudFront al pubblico e ospitato tramite un URL personalizzato simile al seguente:

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

   Sostituisci il *`nome di dominio alternativo di CloudFront distribuzione`* e il *`percorso del video S3`* con i valori appropriati. Un esempio di URL è **`https://www.example.com/sample.mp4`**.

(Facoltativo) Passaggio 8: Visualizza i dati sulle richieste ricevute dalla tua distribuzione CloudFront

Per visualizzare i dati sulle richieste ricevute dalla tua CloudFront distribuzione

1. Accedi a AWS Management Console e apri la CloudFront console all'indirizzohttps://console.aws.amazon.com/cloudfront/v4/home.

2. Nel pannello di navigazione sulla sinistra, in Report e analisi dei dati, scegli i report dalla console, che vanno da Statistiche sulla cache, Oggetti popolari, Referrer principali, Utilizzo e Visualizzatori.

Puoi filtrare il pannello di controllo di ogni report. Per ulteriori informazioni, consulta CloudFront Report in the Console nella Amazon CloudFront Developer Guide.

3. Per filtrare i dati, scegli l'ID della CloudFront distribuzione che hai creato nel passaggio 4.

Fase 9: Pulizia

Se hai ospitato un video in streaming su S3 utilizzando CloudFront Route 53 solo come esercizio di apprendimento, elimina le AWS risorse che hai allocato in modo da non incorrere in costi aggiuntivi.

> **Note**
>
> Quando registri un dominio, il costo è immediato ed è irreversibile. Puoi scegliere di non rinnovare automaticamente il dominio, ma il pagamento è anticipato e resti proprietario per un anno. Per maggiori informazioni, consulta Registrazione di un nuovo dominio nella Guida per gli sviluppatori di Amazon Route 53.

Fasi secondarie

- Elimina la distribuzione CloudFront
- Eliminazione del registro DNS
- Eliminazione della zona ospitata pubblica per il dominio personalizzato
- Eliminazione del nome di dominio personalizzato da Route 53
- Eliminazione del video originale nel bucket S3 di origine
- Eliminazione del bucket S3 di origine

Elimina la distribuzione CloudFront

1. Accedi a AWS Management Console e apri la CloudFront console all'indirizzohttps://console.aws.amazon.com/cloudfront/v4/home.
2. Nel riquadro di navigazione a sinistra, scegliereDistribuzioni.
3. Nella colonna Origins, trova la CloudFront distribuzione corretta cercando il nome di origine, che inizia con il nome del bucket S3 per il bucket che hai creato nel passaggio 1 (ad esempio,). **`tutorial-bucket`**
4. Per eliminare la CloudFront distribuzione, devi prima disabilitarla.

- Se il valore della colonna Stato è Abilitato e il valore di Ultima modifica è il timestamp dell'ultima modifica della distribuzione, procedi a disabilitare la distribuzione prima di eliminarla.
- Se il valore di Stato è Abilitato e il valore di Ultima modifica è Implementazione in corso, attendi fino a quando Stato passa alla marca temporale dell'ultima modifica della distribuzione. Quindi procedi a disabilitare la distribuzione prima di eliminarla.

5. Per disabilitare la CloudFront distribuzione, procedi come segue:

   a. NellaDistribuzioniSelezionare la casella di controllo accanto all'ID della distribuzione che si desidera eliminare.

   b. Per disabilitare la distribuzione, scegli Disabilita (e poi di nuovo Disabilita per confermare.

      Se disabiliti una distribuzione a cui è associato un nome di dominio alternativo, CloudFront smette di accettare il traffico per quel nome di dominio (ad esempio`www.example.com`), anche se un'altra distribuzione ha un nome di dominio alternativo con un carattere jolly (`*`) che corrisponde allo stesso dominio (ad esempio). `*.example.com`

   c. Il valore di Status (Stato) cambia immediatamente in Disabled (Disabilitato). Attendere fino a quando il valore di Ultima modifica passa da Implementazione in corso al timestamp dell'ultima modifica della distribuzione.

      Poiché è CloudFront necessario propagare questa modifica a tutte le edge location, potrebbero essere necessari alcuni minuti prima che l'aggiornamento sia completo e che sia disponibile l'opzione Elimina per eliminare la distribuzione.

6. Per eliminare la distribuzione disabilitata, procedi come indicato di seguito:

   a. Selezionare la casella di controllo accanto all'ID della distribuzione che si desidera eliminare.

   b. Scegli Elimina e seleziona Elimina per confermare.

Eliminazione del registro DNS

Se si desidera eliminare la zona ospitata pubblica per il dominio (incluso il record DNS), vedereEliminazione della zona ospitata pubblica per il dominio personalizzatonellaGuida per sviluppatori di Amazon Route 53. Se desideri solo eliminare il registro DNS creato nella Fase 6, procedi come segue:

1. Accedi AWS Management Console e apri la console Route 53 all'indirizzo https://console.aws.amazon.com/route53/.
2. Nel riquadro di navigazione a sinistra, scegliere Hosted zones (Zone ospitate).
3. Nella pagina Zone ospitate seleziona il nome della zona ospitata creata da Route 53 in Prerequisiti (ad esempio, **`example.com`**).
4. Nell'elenco dei registri, seleziona quelli che desideri eliminare (i registri creati nella Fase 6).

   > **Note**
   >
   > Non è possibile eliminare i record con un valore di Tipo pari a NS o SOA.

5. Seleziona Delete records (Elimina registri).
6. Per confermare l'eliminazione, scegliere Delete (Elimina).

   Le modifiche ai registri richiedono tempo per propagarsi ai server DNS di Route 53. Attualmente, l'unico modo per verificare che le modifiche si siano propagate è utilizzare l'azione GetChange API. In genere le modifiche si propagano a tutti i server Route 53 entro 60 secondi.

Eliminazione della zona ospitata pubblica per il dominio personalizzato

> **Warning**
>
> Se desideri mantenere la registrazione del tuo dominio ma interrompere il routing del traffico Internet sul tuo sito o applicazione Web, ti consigliamo di eliminare i registri nella zona ospitata (come sopra) invece di eliminare la zona ospitata.
> Inoltre, se elimini una zona ospitata, qualcuno potrebbe utilizzare il dominio e instradare il traffico verso le proprie risorse utilizzando il tuo nome di dominio.
> Se elimini una zona ospitata, non puoi annullarne l'eliminazione. Devi creare una nuova zona ospitata e aggiornare i server di nomi per la registrazione del tuo dominio, operazione che può richiedere fino a 48 ore per rendere effettiva la modifica.
> Se desideri rendere il dominio non disponibile su Internet, per prima cosa puoi trasferire il servizio DNS su un servizio DNS gratuito e quindi eliminare la zona ospitata di Route 53. In questo modo si impedisce che query DNS future vengano instradate in modo non corretto.
>
> 1. Se il dominio è registrato con Route 53, consulta Aggiunta o modifica di server di nomi e glue record per un dominio nella Guida per gli sviluppatori di Amazon Route 53 per

informazioni su come sostituire i server di nomi di Route 53 con i server di nomi per il nuovo servizio DNS.

2. Se il dominio è registrato con un altro registrar, utilizza il metodo fornito dal registrar per modificare i server di nomi per il dominio.

> **Note**
>
> Se stai eliminando una zona ospitata per un sottodominio (`www.example.com`), non devi modificare i server di nomi per il dominio (`example.com`).

1. Accedi AWS Management Console e apri la console Route 53 all'indirizzo https://console.aws.amazon.com/route53/.
2. Nel riquadro di navigazione a sinistra, scegliere Hosted zones (Zone ospitate).
3. Nella pagina Hosted zones (Zone ospitate), scegli il nome della zona ospitata che desideri eliminare.
4. Nella scheda Records (Registri) della zona ospitata, conferma che la zona ospitata che desideri eliminare contiene solo un registro NS e uno SOA.

   Se contiene registri aggiuntivi, eliminali.

   Se hai creato record NS per sottodomini nella zona ospitata, elimina anche questi.
5. Nella scheda DNSSEC signing (Firma DNSSEC) per la zona ospitata, disabilita la firma DNNSSEC, se abilitata. Per ulteriori informazioni, consulta Registrazione delle query DNS nella Guida per sviluppatori di Amazon Route 53.
6. Nella parte superiore della pagina dei dettagli della zona ospitata, scegliereElimina zona.
7. Per confermare l'eliminazione immetti **`delete`**, quindi scegli Elimina.

Eliminazione del nome di dominio personalizzato da Route 53

Per la maggior parte dei domini di primo livello (TLD), è possibile eliminare la registrazione, se non è più necessaria. Se elimini la registrazione di un nome di dominio da Route 53 prima della scadenza prevista della registrazione, la quota di registrazione AWS non viene rimborsata. Per maggiori informazioni, consulta Eliminazione della registrazione di un nome di dominio nella Guida per gli sviluppatori di Amazon Route 53.

> **Important**
>
> Se desideri trasferire il dominio da un altro registrar Account AWS o trasferirlo a un altro registrar, non eliminare il dominio e aspettati di registrarlo nuovamente immediatamente. Al contrario, consulta la documentazione relativo nella sezioneGuida per sviluppatori di Amazon Route 53:
>
> - Trasferimento di un dominio a un altro Account AWS
> - Trasferimento di un dominio da Amazon Route 53 a un altro registrar

Eliminazione del video originale nel bucket S3 di origine

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Nome bucket scegli il nome del bucket in cui hai caricato il video originale nella Fase 2 (ad esempio, **`tutorial-bucket`**).
4. Nella scheda Oggetti, seleziona la casella di controllo a sinistra del nome dell'oggetto da eliminare (ad esempio, `sample.mp4`).
5. Scegli Elimina.
6. UNDEREliminare permanentemente gli oggetti?, immettere**`permanently delete`**per confermare di voler eliminare questo oggetto.
7. Scegliere Delete objects (Elimina oggetti).

Eliminazione del bucket S3 di origine

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket scegli il pulsante di opzione accanto al nome del bucket creato nella Fase 1(ad esempio, **`tutorial-bucket`**).
4. Scegli Elimina.
5. Nella pagina Delete bucket (Elimina bucket) conferma che desideri eliminare il bucket inserendone il nome nel campo di testo e quindi scegli Delete bucket (Elimina bucket).

Passaggi successivi

Dopo aver completato questo tutorial, puoi esplorare altri casi d'uso correlati:

- Transcodifica i video S3 nei formati di streaming necessari a un particolare televisore o dispositivo connesso prima di ospitarli su una distribuzione. CloudFront

  Per utilizzare Amazon S3 Batch Operations AWS Lambda e AWS Elemental MediaConvert transcodificare in batch una raccolta di video in una varietà di formati multimediali di output, consulta. Tutorial: transcodifica in batch di video con S3 Batch Operations e AWS LambdaAWS Elemental MediaConvert
- Ospita altri oggetti archiviati in S3, come immagini, audio, grafica animata, fogli di stile, HTML JavaScript, app React e così via, utilizzando Route 53. CloudFront

  Per un esempio, consulta Tutorial: Configurazione di un sito Web statico utilizzando un dominio personalizzato registrato con Route 53 e Velocizza il tuo sito Web con Amazon CloudFront.
- Utilizza Amazon S3 Transfer Acceleration per configurare trasferimenti di file veloci e sicuri. Transfer Acceleration può velocizzare il caricamento dei video nel bucket S3 per il trasferimento a lunga distanza di video di grandi dimensioni. Transfer Acceleration migliora le prestazioni di trasferimento instradando il traffico attraverso le edge location distribuite CloudFront a livello globale e sulle reti dorsali. AWS Utilizza anche ottimizzazioni del protocollo di rete. Per ulteriori informazioni, consulta Configurazione di trasferimenti veloci e sicuri di file con Amazon S3 Transfer Acceleration.

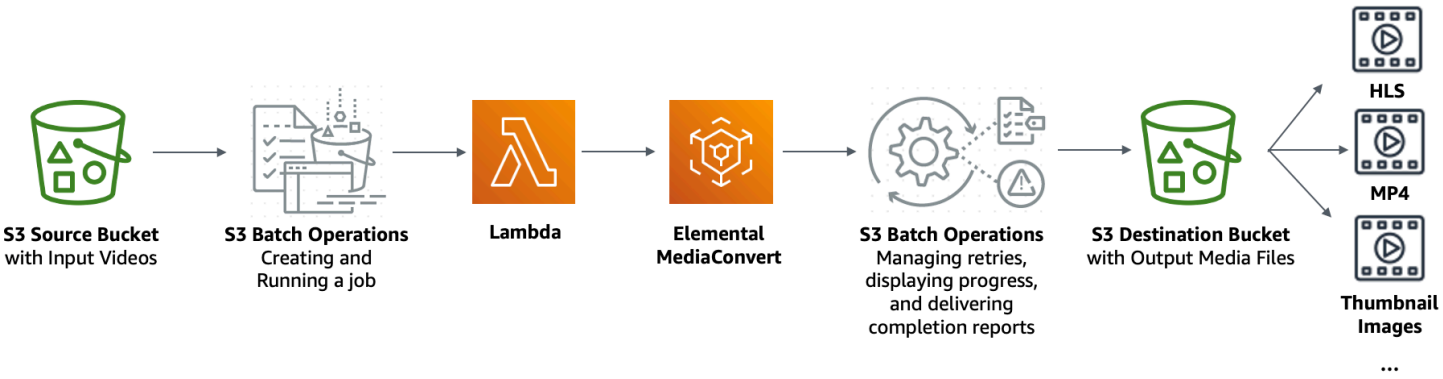
Tutorial: transcodifica in batch di video con S3 Batch Operations e AWS LambdaAWS Elemental MediaConvert

I consumatori di video utilizzano dispositivi di tutte le forme, dimensioni ed età per godere dei contenuti multimediali. Questa vasta gamma di dispositivi rappresenta una sfida per i creatori e i distributori di contenuti. Invece di essere in un one-size-fits-all formato, i video devono essere convertiti in modo che possano coprire un'ampia gamma di dimensioni, formati e bitrate. Questa operazione di conversione è ancora più difficile quando si dispone di un numero elevato di video che devono essere convertiti.

AWS offre un metodo per creare un'architettura scalabile e distribuita che esegue le seguenti operazioni:

- Importa video di input
- Elabora i video per la riproduzione su un'ampia gamma di dispositivi
- Memorizza i file multimediali transcodificati
- Fornisce i file multimediali di output per soddisfare la domanda

Quando hai repository video di grandi dimensioni archiviati in Amazon S3, puoi transcodificare questi video dal loro formato di origine in più tipi di file nelle dimensioni, nella risoluzione o nel formato necessario per un determinato lettore video o dispositivo. In particolare, S3 Batch Operations offre una soluzione per richiamare AWS Lambda funzioni per i video di input esistenti in un bucket sorgente S3. Quindi, le funzioni Lambda richiamano AWS Elemental MediaConvert perché esegua processi di transcodifica di video su larga scala. I file multimediali di output convertiti sono archiviati in un bucket di destinazione S3.



Obiettivo

In questa esercitazione imparerai a impostare le operazioni in batch S3 per richiamare una funzione Lambda per la transcodificazione in batch di video memorizzati in un bucket S3 di origine. La funzione Lambda chiama MediaConvert per transcodificare i video. Gli output per ogni video nel bucket S3 di origine hanno le caratteristiche mostrate di seguito:

- Un flusso di bitrate adattivo HTTP Live Streaming (HLS) per la riproduzione su dispositivi di dimensioni multiple e con larghezza di banda variabile.
- Un file video MP4
- Immagini in miniatura raccolte a intervalli

Argomenti

- Prerequisiti

Prerequisiti

Prima di iniziare questo tutorial, devi disporre di un bucket Amazon S3 di origine (ad esempio, **`tutorial-bucket-1`**) dove siano già archiviati video da transcodificare.

Se lo desideri, puoi assegnare al bucket un altro nome. Per ulteriori informazioni sulle regole di denominazione dei bucket in Amazon S3, consulta Regole di denominazione dei bucket.

Per il bucket S3 di origine, mantieni le impostazioni relative a Impostazioni di blocco dell'accesso pubblico per questo bucket sui valori di default (l'opzione Blocca tutto l'accesso pubblico è abilitata). Per ulteriori informazioni, consulta Creazione di un bucket.

Per ulteriori informazioni sul caricamento di video nel bucket S3 di origine, consulta Caricamento degli oggetti. Quando carichi un video in S3, puoi anche possibile utilizzare Amazon S3 Transfer Acceleration per configurare trasferimenti di file veloci e sicuri. Transfer Acceleration può velocizzare il caricamento dei video nel bucket S3 per il trasferimento a lunga distanza di video di grandi dimensioni. Per ulteriori informazioni, consulta Configurazione di trasferimenti veloci e sicuri di file con Amazon S3 Transfer Acceleration.

Fase 1: Creazione di un bucket S3 per i file multimediali di output

In questa fase, viene creato un bucket S3 di destinazione per archiviare i file multimediali di output convertiti. Puoi inoltre creare una configurazione per l'abilitazione della condivisione di risorse multiorigine (CORS, Cross Origin Resource Sharing) per permettere l'accesso tra origini ai file multimediali transcodificati archiviati nel bucket S3 di destinazione.

Fasi secondarie

- Creazione di un bucket per i file multimediali di output
- Aggiunta di una configurazione CORS a un bucket S3 di output

Creazione di un bucket per i file multimediali di output

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Seleziona Crea bucket.
4. Per Nome bucket, specifica un nome per il bucket, ad esempio **`tutorial-bucket-2`**.
5. Per Regione, scegli Regione AWS dove vuoi che risieda il bucket.
6. Per garantire l'accesso pubblico ai file multimediali di output, in Block Public Access settings for this bucket (Impostazioni di blocco dell'accesso pubblico per questo bucket), deseleziona Block all public access (Blocca tutto l'accesso pubblico).

   > **Warning**
   >
   > Prima di completare questa fase, consulta Blocco dell'accesso pubblico allo storage Amazon S3 per confermare di avere compreso e accettato i rischi connessi alla concessione di un accesso pubblico. Quando si disattivano le impostazioni di blocco dell'accesso pubblico per rendere pubblico il bucket, chiunque su Internet può accedere al bucket. Consigliamo di bloccare tutti gli accessi pubblici ai bucket.
   > Se non desideri cancellare le impostazioni Block Public Access, puoi utilizzare Amazon CloudFront per distribuire i file multimediali transcodificati agli spettatori (utenti finali). Per ulteriori informazioni, consulta Tutorial: hosting di video in streaming su richiesta con Amazon S3, Amazon e CloudFront Amazon Route 53.

7. Seleziona la casella di controllo accanto a I acknowledge that the current settings might result in this bucket and the objects within becoming public. (Riconosco che le impostazioni correnti possono portare il bucket e gli oggetti all'interno a diventare pubblici).
8. Mantieni le impostazioni rimanenti impostate sui valori predefiniti.
9. Seleziona Crea bucket.

Aggiunta di una configurazione CORS a un bucket S3 di output

Una configurazione JSON CORS definisce un metodo con cui le applicazioni Web client (lettori video in questo contesto) caricate in un dominio possono riprodurre file multimediali di output transcodificati in un dominio differente.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket scegli il nome del bucket creato in precedenza (ad esempio, **`tutorial-bucket-2`**).
4. Scegli la scheda Autorizzazioni.
5. Nella sezione Cross-Origin Resource Sharing (CORS) scegliere Edit (Modifica).
6. Nella casella di testo della configurazione CORS, copia e incolla la nuova configurazione CORS mostrata di seguito.

   La configurazione CORS deve essere in formato JSON. In questo esempio, l'attributo `AllowedOrigins` utilizza il carattere jolly (*) per specificare tutte le origini. Se conosci la tua origine specifica, puoi limitare l'attributo `AllowedOrigins` all'URL del lettore specifico. Per ulteriori informazioni sulla configurazione di questo e altri attributi, consulta Configurazione CORS.

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

       }
   ]
   ```

7. Seleziona Salvataggio delle modifiche.

Fase 2: Creare un ruolo IAM per MediaConvert

Per poter AWS Elemental MediaConvert transcodificare i video di input archiviati nel bucket S3, devi avere un ruolo di servizio AWS Identity and Access Management (IAM) che conceda MediaConvert le autorizzazioni per leggere e scrivere file video da e verso i bucket di origine e destinazione S3. Quando esegui lavori di transcodifica, la console utilizza questo ruolo. MediaConvert

Per creare un ruolo IAM per MediaConvert

1. Crea un ruolo IAM scegliendo un nome di ruolo (ad esempio, **`tutorial-mediaconvert-role`**). Per creare questo ruolo, segui i passaggi in Crea il tuo MediaConvert ruolo in IAM (console) nella Guida per l'AWS Elemental MediaConvert utente.
2. Dopo aver creato il ruolo IAM per MediaConvert, nell'elenco dei ruoli, scegli il nome del ruolo per MediaConvert cui hai creato (ad esempio,**`tutorial-mediaconvert-role`**).
3. Nella pagina Riepilogo, copia l'ARN ruolo (che inizia con `arn:aws:iam::`) e salva l'ARN per utilizzarlo in un secondo momento.

   Per ulteriori informazioni sull'utilizzo degli ARN, consulta Amazon Resource Name (ARN) in Riferimenti generali di AWS .

Fase 3: creazione di un ruolo IAM per la funzione Lambda

Per transcodificare in batch i video con e MediaConvert S3 Batch Operations, usi una funzione Lambda per connettere questi due servizi e convertire i video. Questa funzione Lambda deve avere un ruolo IAM che conceda alla funzione Lambda le autorizzazioni di accesso e le operazioni in batch di S3. MediaConvert

Fasi secondarie

- Creazione di un ruolo IAM per la funzione Lambda
- Incorporazione di una policy inline per il ruolo IAM della funzione Lambda

Creazione di un ruolo IAM per la funzione Lambda

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra, scegli Ruoli, quindi Crea ruolo.

3. Scegli il tipo di ruolo Servizio AWS dopodiché in Casi d'uso comuni scegli Lambda.
4. Scegli Successivo: autorizzazioni.
5. Sulla pagina Collega policy di autorizzazioni, immetti **`AWSLambdaBasicExecutionRole`** nella casella Filtra policy. Per allegare la policy gestita AWSLambdaBasicExecutionRolea questo ruolo per concedere le autorizzazioni di scrittura ad Amazon CloudWatch Logs, seleziona la casella di controllo accanto a. AWSLambdaBasicExecutionRole
6. Scegli Successivo: Tag.
7. (Facoltativo) Aggiungi i tag alla policy gestita.
8. Scegli Prossimo: Rivedi.
9. Per Nome ruolo, inserisci **`tutorial-lambda-transcode-role`**.
10. Scegli Crea ruolo.

Incorporazione di una policy inline per il ruolo IAM della funzione Lambda

Per concedere le autorizzazioni alla MediaConvert risorsa necessaria per l'esecuzione della funzione Lambda, è necessario utilizzare una policy in linea.

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra, seleziona Ruoli.
3. Nell'elenco Ruoli scegli il nome del ruolo IAM creato in precedenza per la funzione Lambda (ad esempio, **`tutorial-lambda-transcode-role`**).
4. Scegliere la scheda Permissions (Autorizzazioni).
5. Scegliere Add inline policy (Aggiungi policy inline).
6. Scegli la scheda JSON e copia e incolla la seguente policy JSON.

   Nella policy JSON, sostituisci il valore ARN di esempio `Resource` di con il ruolo ARN del ruolo IAM MediaConvert per il quale hai creato nella Fase 2 (ad esempio,). **`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. Scegliere Review policy (Esamina policy).
8. In Nome, inserisci **`tutorial-lambda-policy`**.
9. Scegliere Create Policy (Crea policy).

Una volta creata, la policy inline viene automaticamente incorporata nel ruolo IAM della funzione Lambda.

Fase 4: Creazione di una funzione Lambda per la transcodifica dei video

In questa sezione del tutorial, crei una funzione Lambda utilizzando l'SDK per Python per l'integrazione con S3 Batch Operations e. MediaConvert Per iniziare a transcodificare i video già archiviati nel bucket S3 di origine, esegui un processo di operazioni in batch S3 che richiama direttamente la funzione Lambda per ogni video nel bucket S3 di origine. Quindi, la funzione Lambda invia un processo di transcodifica per ogni video a. MediaConvert

Fasi secondarie

- Scrittura del codice della funzione Lambda e creazione di un pacchetto di implementazione
- Creare una funzione Lambda con un ruolo di esecuzione (console)
- Implementa la tua funzione Lambda con gli archivi in file .zip e configura la funzione Lambda (console)

Scrittura del codice della funzione Lambda e creazione di un pacchetto di implementazione

1. Nel computer locale, crea un cartella denominata `batch-transcode`.
2. Nella cartella `batch-transcode`, crea un file con le impostazioni del processo JSON. Ad esempio, è possibile utilizzare le impostazioni fornite in questa sezione e denominare il file `job.json`.

   Un file `job.json` specifica le seguenti informazioni:

   - Quali file transcodificare
   - Come transcodificare i tuoi video di input
   - Quali file multimediali di output si desidera creare
   - Come denominare i file transcodificati
   - Dove salvare i file transcodificati
   - Le funzioni avanzate da applicare e così via.

In questo tutorial utilizziamo il seguente file `job.json` per creare i seguenti output per ogni video del bucket S3 di origine:

- Un flusso di bitrate adattivo HTTP Live Streaming (HLS) per la riproduzione su dispositivi di dimensioni differenti e con larghezza di banda variabile.
- Un file video MP4
- Immagini in miniatura raccolte a intervalli

Questo file `job.json` di esempio utilizza il bitrate della variabile definita dalla qualità (QVCR, Quality-Defined Variable Bitrate) per ottimizzare la qualità del video. L'output HTTP Live Streaming (HLS) è conforme a Apple (audio non mixato al video, durata del segmento corretta di 6 secondi e qualità video ottimizzata tramite QVBR automatico).

Se non si desidera utilizzare le impostazioni di esempio fornite qui, è possibile generare una specifica `job.json` basata sul proprio caso d'uso. Per garantire la coerenza tra gli output, assicurati che i file di input abbiano configurazioni video e audio simili. Per tutti i file di input con configurazioni video e audio diverse, è consigliabile creare automazioni separate (impostazioni `job.json` univoche). Per ulteriori informazioni, consulta Example AWS Elemental MediaConvert job settings in JSON nella Guida per l'utente di 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. Nella cartella `batch-transcode`, crea un file con una funzione Lambda. Puoi utilizzare il seguente esempio Python e denominare il file `convert.py`.

   Le operazioni in batch S3 inviano dati di processi specifici a una funzione Lambda e richiedono i dati dei risultati. Per gli esempi di richiesta e risposta per la funzione Lambda, le informazioni sui codici di risposta e dei risultati e le funzioni Lambda di esempio per le operazioni in batch S3, consulta Funzione Invoke 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. Per creare un pacchetto di implementazione con `convert.py` e `job.json` come file `.zip` denominato `lambda.zip`, nel computer locale, apri la cartella `batch-transcode` creata in precedenza ed emetti il comando seguente.

   Per utenti macOS, eseguire il seguente comando:

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

   Per utenti Windows, esegui questi comandi:

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

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

Creare una funzione Lambda con un ruolo di esecuzione (console)

1. Apri la console all'indirizzo https://console.aws.amazon.com/lambda/ AWS Lambda .

2. Nel pannello di navigazione a sinistra, scegli Functions (Funzioni).
3. Selezionare Create function (Crea funzione).
4. Scegli Author from scratch (Crea da zero).
5. In Basic information (Informazioni di base) eseguire queste operazioni:

   a. Nel campo Function name (Nome funzione), immettere **`tutorial-lambda-convert`**.

   b. In Runtime, scegli Python 3.8 o una versione successiva.
6. Scegli Change default execution role (Cambia ruolo di esecuzione predefinito) e in Execution role (Ruolo di esecuzione) scegli Use an existing role (Utilizza un ruolo esistente).
7. In Ruolo esistente, scegli il nome del ruolo IAM creato per la funzione Lambda nella Fase 3 (ad esempio, **`tutorial-lambda-transcode-role`**).
8. Mantieni le impostazioni rimanenti impostate sui valori di default.
9. Scegli Crea funzione.

Implementa la tua funzione Lambda con gli archivi in file .zip e configura la funzione Lambda (console)

1. Nel riquadro Origine codice della pagina della funzione Lambda creata in precedenza (ad esempio, **`tutorial-lambda-convert`**), scegli Carica da e poi File .zip.
2. Scegli Upload (Carica) per selezionare il file `.zip` locale.
3. Scegli il file `lambda.zip` creato in precedenza, quindi scegli Apri.
4. Selezionare Salva.
5. Nel pannello Runtime settings (Impostazioni runtime), scegli Edit (Modifica).
6. Per indicare al runtime Lambda quale metodo gestore richiamare nel codice della funzione Lambda, inserisci **`convert.handler`** nel campo Gestore.

   Quando si configura una funzione in Python, il valore dell'impostazione del gestore è costituito dal nome del file e dal nome del modulo del gestore, separati da un punto (`.`). Ad esempio, `convert.handler` richiama il metodo `handler` definito nel file `convert.py`.
7. Selezionare Salva.
8. Nella pagina della funzione Lambda, scegli la scheda Configuration (Configurazione). Nel pannello di navigazione a sinistra nella scheda Configurazione, scegli Variabili di ambiente, quindi scegli Modifica.

9. Scegli Add environment variable (Aggiungi variabile d'ambiente). Quindi, inserisci Chiave e Valore per ciascuna delle variabili di ambiente elencate di seguito.

   - Chiave: **`DestinationBucket`** Valore: **`tutorial-bucket-2`**

     Questo valore è il bucket S3 per i file multimediali di output creati nella Fase 1.

   - Chiave: **`MediaConvertRole`** Valore: **`arn:aws:iam::`*`111122223333`*`:role/tutorial-mediaconvert-role`**

     Questo valore è l'ARN del ruolo IAM per MediaConvert il quale hai creato nel passaggio 2. Assicurati di sostituire questo ARN con l'ARN effettivo del tuo ruolo IAM.

   - Chiave: **`Application`** Valore: **`Batch-Transcoding`**

     Questo valore è il nome dell'applicazione.

10. Selezionare Salva.
11. (Facoltativo) Nella scheda Configuration (Configurazione), nella sezione General configuration (Configurazione generale) del pannello di navigazione a sinistra, seleziona Edit (Modifica). Nel campo Timeout inserisci **`2`** min **`0`** sec. Quindi, scegliere Save (Salva).

    Il Timeout è la quantità di tempo consentita da Lambda per l'esecuzione di una funzione per una chiamata prima di arrestarla. Il valore predefinito è 3 secondi. I prezzi si basano sulla quantità di memoria configurata e sulla quantità di tempo di esecuzione del codice. Per ulteriori informazioni, consultare Prezzi di AWS Lambda.

Fase 5: Configurazione dell'inventario Amazon S3 per il bucket S3 di origine

Dopo aver impostato la funzione di transcodifica Lambda, devi creare un processo di operazioni in batch S3 per transcodificare una serie di video. Innanzitutto, devi disporre di un elenco degli oggetti video di input sui quali desideri che le operazioni in Batch S3 eseguano l'azione di transcodifica specificata. Per ottenere un elenco di oggetti video di input, puoi generare un report di inventario S3 per il bucket S3 di origine (ad esempio, **`tutorial-bucket-1`**).

Fasi secondarie

- Creazione e configurazione di un bucket per i report di inventario S3 dei video di input
- Configurazione dell'inventario Amazon S3 per il bucket S3 di origine dei video
- Controllo del report di inventario per il bucket S3 di origine dei video

Creazione e configurazione di un bucket per i report di inventario S3 dei video di input

Per archiviare un report di inventario S3 che elenca gli oggetti del bucket S3 di origine, devi creare un bucket di destinazione dell'inventario S3 e configurare una policy perché il bucket possa scrivere i file di inventario nel bucket S3 di origine.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Seleziona Crea bucket.
4. Per Nome bucket, specifica un nome per il bucket, ad esempio **`tutorial-bucket-3`**.
5. Per Regione AWS, scegli Regione AWS dove vuoi che risieda il bucket.

   Il bucket di destinazione dell'inventario deve trovarsi nello stesso Regione AWS del bucket di origine in cui stai configurando S3 Inventory. Il bucket di destinazione dell'inventario può trovarsi in un diverso Account AWS.
6. In Impostazioni di blocco dell'accesso pubblico per questo bucket, mantieni le impostazioni di default (l'opzione Blocca tutto l'accesso pubblico è abilitata).
7. Mantieni le impostazioni rimanenti impostate sui valori di default.
8. Seleziona Crea bucket.
9. Nell'elenco Buckets (Bucket) scegli il nome del bucket appena creato (ad esempio, **`tutorial-bucket-3`**).
10. Per concedere ad Amazon S3 l'autorizzazione a scrivere dati per i report di inventario nel bucket di destinazione dell'inventario S3, seleziona la scheda Autorizzazioni.
11. Scorri verso il basso fino alla sezione Policy di bucket e scegli Modifica. Viene visualizzata la pagina Policy del bucket.
12. Per concedere le autorizzazioni per l'inventario S3, nel campo Policy copia la seguente policy del bucket.

    Sostituisci i tre valori di esempio rispettivamente con i seguenti valori:

    - Il nome del bucket creato per archiviare i rapporti di inventario (ad esempio, *`tutorial-bucket-3`*).
    - Il nome del bucket di origine che archivia i video di input (ad esempio, *`tutorial-bucket-1`*).

- L' Account AWS ID che hai usato per creare il bucket di sorgenti video S3 (ad esempio,. *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. Seleziona Salvataggio delle modifiche.

Configurazione dell'inventario Amazon S3 per il bucket S3 di origine dei video

Per generare un elenco di file flat di oggetti video e metadati, devi configurare l'inventario S3 per il bucket S3 di origine dei video. Questi report pianificati possono includere tutti gli oggetti del bucket oppure oggetti raggruppati in base a un prefisso condiviso. In questo tutorial, il report di inventario S3 include tutti gli oggetti video nel bucket S3 di origine.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Per configurare un report di inventario S3 dei video di input nel bucket S3 di origine, nell'elenco Bucket scegli il nome del bucket S3 di origine (ad esempio, **`tutorial-bucket-1`**).

4. Scegliere la scheda Management (Gestione),
5. Scorri fino alla sezione Configurazione dell'inventario e scegli Crea configurazione dell'inventario.
6. Nel campo Inventory configuration name (Nome configurazione inventario) inserisci un nome (ad esempio, **`tutorial-inventory-config`**).
7. In Ambito inventario, scegli Solo versione corrente per Versioni oggetto e mantieni le altre impostazioni di Ambito inventario sui valori di default per questo tutorial.
8. Nella sezione Dettagli report, per Bucket di destinazione, scegli Questo account.
9. In Destinazione, scegli Sfoglia S3 e scegli il bucket di destinazione creato in precedenza per salvare i report di inventario (ad esempio, **`tutorial-bucket-3`**). Quindi, scegli Seleziona percorso.

   Il bucket di destinazione dell'inventario deve trovarsi nello stesso bucket Regione AWS di origine in cui stai configurando S3 Inventory. Il bucket di destinazione dell'inventario può trovarsi in un diverso Account AWS.

   Nel campo Destination bucket (Bucket di destinazione) la Destination bucket permission (Autorizzazione per il bucket di destinazione) viene aggiunta alla policy del bucket di destinazione dell'inventario per consentire ad Amazon S3 di inserirvi i dati. Per ulteriori informazioni, consulta Creazione di una policy di bucket di destinazione.
10. In Frequenza, seleziona Giornaliero.
11. Per Formato di output, seleziona CSV.
12. In Stato, scegli Abilitato.
13. In Crittografia lato server, scegli Disabilita per questo tutorial.

    Per ulteriori informazioni, consulta Configurazione dell'inventario utilizzando la console S3 e Concessione ad Amazon S3 dell'autorizzazione per l'utilizzo della chiave gestita dal cliente per la crittografia.
14. Nella sezione Campi aggiuntivi - facoltativo, seleziona Dimensioni, Ultima modifica e Classe di archiviazione.
15. Scegli Create (Crea).

Per ulteriori informazioni, consulta Configurazione dell'inventario utilizzando la console S3.

Controllo del report di inventario per il bucket S3 di origine dei video

Quando viene pubblicato un elenco di inventario, i file manifesto vengono inviati al bucket di destinazione dell'inventario S3.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckets (Bucket) scegli il nome del bucket di origine dei video (ad esempio, **`tutorial-bucket-1`**).
4. Seleziona Gestione.
5. Per verificare se il report di inventario S3 è pronto per la creazione di un processo di operazioni in batch S3 nella Fase 7, in Configurazioni di inventario, verifica se il pulsante Create processo dal manifesto è abilitato.

   > **Note**
   >
   > La consegna del primo report di inventario può richiedere fino a 48 ore. Se il pulsante Create job from manifest (Crea processo dal manifesto) è disattivato, il primo report di inventario non è stato consegnato. Devi attendere che venga consegnato il primo report di inventario e che il pulsante Crea processo dal manifesto sia abilitato per creare un processo di operazioni in batch S3 nella Fase 7.

6. Per controllare un report di inventario S3 (`manifest.json`), nella colonna Destinazione, scegli il nome del bucket di destinazione dell'inventario creato in precedenza per l'archiviazione dei report di inventario (ad esempio, **`tutorial-bucket-3`**).
7. Nella scheda Oggetti, scegli la cartella esistente con il nome del bucket di origine S3 (ad esempio, **`tutorial-bucket-1`**). Quindi scegli il nome che hai inserito in Nome configurazione inventario quando hai creato la configurazione dell'inventario (ad esempio, **`tutorial-inventory-config`**).

   Puoi visualizzare un elenco di cartelle denominate con la data di generazione dei report.
8. Per controllare il report di inventario S3 giornaliero di una certa data, scegli una cartella denominata con una data di generazione, quindi scegli `manifest.json`.
9. Per controllare i dettagli del report di inventario in una data specifica, nella pagina manifest.json, scegli Download (Scarica) o Open (Apri).

Fase 6: creazione di un ruolo IAM per le operazioni in batch S3

Per utilizzare le operazioni in batch S3 per eseguire la transcodifica in batch, per prima cosa devi creare un ruolo IAM per consentire ad Amazon S3 di disporre delle autorizzazioni per eseguire le operazioni in batch S3.

Fasi secondarie

- Creazione di una policy IAM per le operazioni in batch S3
- Creare un ruolo IAM per le operazioni in batch S3 e assegna le policy di autorizzazione

Creazione di una policy IAM per le operazioni in batch S3

Devi creare una policy IAM che fornisca alle operazioni in batch S3 l'autorizzazione per leggere il manifesto di input, richiamare la funzione Lambda e scrivere il report di completamento del processo di operazioni in batch S3.

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra, seleziona Policy.
3. Scegli Crea policy.
4. Scegliere la scheda JSON.
5. Nel campo di testo JSON incolla la seguente policy JSON.

   Nella policy JSON, sostituisci i quattro valori di esempio con i seguenti valori:

   - Il nome del bucket di origine che archivia i video di input (ad esempio, *`tutorial-bucket-1`*).
   - Il nome del bucket di destinazione dell'inventario creato nella Fase 5 per archiviare i file `manifest.json` (ad esempio, *`tutorial-bucket-3`*).
   - Il nome del bucket creato nella Fase 1 per archiviare i file multimediali di output (ad esempio, *`tutorial-bucket-2`*). In questo tutorial, abbiamo messo i report di completamento del processo nel bucket di destinazione per i file multimediali di output.
   - L'ARN del ruolo della funzione Lambda creato nella Fase 4. Per trovare e copiare l'ARN del ruolo della funzione Lambda, completa le seguenti operazioni:
     - In una nuova scheda del browser, apri la pagina Funzioni nella console Lambda all'indirizzo https://console.aws.amazon.com/lambda/home#/functions.

- Nell'elenco Funzioni, scegli la funzione Lambda creata nella Fase 4 (ad esempio, **`tutorial-lambda-convert`**).
- Scegli Copy ARN (Copia 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. Scegliere Next: Tags (Successivo: Tag).
7. Scegliere Next:Review (Successivo: Rivedi).

8. Nel campo Name (Nome), inserire **`tutorial-s3batch-policy`**.
9. Scegli Crea policy.

Creare un ruolo IAM per le operazioni in batch S3 e assegna le policy di autorizzazione

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra, scegli Ruoli, quindi Crea ruolo.
3. Scegli il tipo di ruolo Servizio AWS, quindi seleziona il servizio S3.
4. In Select your use case (Seleziona il tuo caso d'uso), scegli S3 Batch Operations (Operazioni in batch S3).
5. Scegliere Next: Permissions (Successivo: Autorizzazioni).
6. In Collega policy di autorizzazioni, immettere il nome della policy IAM creata in precedenza (ad esempio, **`tutorial-s3batch-policy`**) nella casella di ricerca per filtrare l'elenco di policy. Seleziona la casella di controllo accanto al nome della policy (ad esempio, **`tutorial-s3batch-policy`**).
7. Scegliere Next: Tags (Successivo: Tag).
8. Scegliere Next:Review (Successivo: Rivedi).
9. Per Nome ruolo, inserisci **`tutorial-s3batch-role`**.
10. Scegli Crea ruolo.

    Dopo aver creato il ruolo IAM per le operazioni in batch S3, la seguente policy di attendibilità viene automaticamente associata al ruolo per permettere all'entità del servizio delle operazioni in batch S3 di assumere il ruolo IAM. Questa policy di attendibilità consente al principale del servizio di operazioni in batch S3 di assumere il ruolo IAM.

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

```
    ]
}
```

Fase 7: creazione ed esecuzione di un processo di operazioni in batch S3

Per creare un processo di operazioni in batch S3 per elaborare i video di input nel bucket S3 di origine, devi specificare i parametri per questo particolare processo.

> **Note**
>
> Per iniziare a creare un processo di operazioni in batch S3, devi assicurarti che il pulsante Crea processo dal manifesto sia abilitato. Per ulteriori informazioni, consulta Controllo del report di inventario per il bucket S3 di origine dei video. Se il pulsante Crea processo dal manifesto è disabilitato, il primo report di inventario non è stato consegnato e devi attendere che il pulsante sia abilitato. Dopo aver configurato l'inventario Amazon S3 per il bucket S3 di origine nella Fase 5, la consegna del primo report dell'inventario può richiedere fino a 48 ore.

Fasi secondarie

- Creare un processo di operazioni in batch S3
- Esecuzione del processo di operazioni in batch di S3 per richiamare la funzione Lambda
- (Facoltativo) Controllo del report di completamento
- (Facoltativo) Monitoraggio di ogni chiamata Lambda nella console Lambda
- (Facoltativo) Monitora ogni processo di MediaConvert transcodifica video nella console MediaConvert

Creare un processo di operazioni in batch S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Operazioni in batch.
3. Scegli Crea processo.
4. Per Regione AWS, scegli la regione in cui creare il processo.

In questo tutorial, al fine di utilizzare il processo di operazioni in batch S3 per richiamare una funzione Lambda, devi creare il processo nella stessa regione del bucket S3 di origine dei video in cui si trovano gli oggetti a cui fa riferimento il manifesto.

5. Nella sezione Manifesto, procedi nel seguente modo:

   a. Per Manifest format (Formato manifesto), scegliere S3 inventory report (manifest.json) (Report inventario S3 (manifest.json)).

   b. In Oggetto manifesto scegli Sfoglia S3 per trovare il bucket creato nella Fase 5 per l'archiviazione dei report di inventario (ad esempio, **`tutorial-bucket-3`**). Nella pagina Oggetto manifesto, naviga tra i nomi degli oggetti fino a trovare un file `manifest.json` per una data specifica. Questo file elenca le informazioni su tutti i video che vuoi transcodificare in batch. Una volta trovato il file `manifest.json` da utilizzare, scegli il pulsante di opzione accanto ad esso. Quindi, scegli Seleziona percorso.

   c. (Facoltativo) In ID versione oggetto manifesto - facoltativo, inserisci l'ID versione dell'oggetto manifesto se desideri utilizzare una versione diversa da quella più recente.

6. Seleziona Successivo.

7. Per utilizzare la funzione Lambda per transcodificare tutti gli oggetti elencati nel file `manifest.json` selezionato, in Tipo di operazione, scegli Invoca funzione AWS Lambda .

8. Nella sezione Chiamare una funzione Lambda completa le operazioni seguenti:

   a. Scegli Choose from functions in your account (Scegli tra le funzioni del tuo account).

   b. In Funzione Lambda, scegli la funzione Lambda creata nella Fase 4 (ad esempio, **`tutorial-lambda-convert`**).

   c. In Versione della funzione Lambda mantieni il valore di default $LATEST.

9. Seleziona Successivo. Viene visualizzata la pagina Configura opzioni aggiuntive.

10. In Opzioni aggiuntive mantieni le impostazioni predefinite.

    Per ulteriori informazioni su queste opzioni, consulta Elementi della richiesta di un processo di operazioni in batch.

11. Nella sezione Report di completamento, per Percorso di destinazione del report di completamento, scegli Sfoglia S3. Individua il bucket creato per i file multimediali di output nella Fase 1 (ad esempio, **`tutorial-bucket-2`**). Scegli il pulsante di opzione accanto al nome del bucket. Quindi, scegli Seleziona percorso.

Mantieni le impostazioni rimanenti di Report di completamento impostate sui valori di default. Per ulteriori informazioni sulla configurazione dei report di completamento, consulta Elementi della richiesta di un processo di operazioni in batch. Un report di completamento mantiene un registro dei dettagli del processo e delle operazioni eseguite.

12. Nella sezione Autorizzazioni, seleziona Scegli tra ruoli IAM esistenti. In IAM role (Ruolo IAM) scegli il ruolo IAM per il processo di operazioni in batch S3 creato nella Fase 6 (ad esempio, **`tutorial-s3batch-role`**).
13. Seleziona Successivo.
14. Nella pagina Revisione, rivedi le impostazioni. Quindi seleziona Crea processo.

    Dopo che S3 ha terminato la lettura del manifesto del processo di operazioni in batch S3, il processo imposta lo stato del processo su In attesa di conferma dell'esecuzione. Per visualizzare gli aggiornamenti dello stato del processo, aggiorna la pagina. Non sarà possibile eseguire il processo finché lo stato non sarà In attesa di esecuzione della conferma.

Esecuzione del processo di operazioni in batch di S3 per richiamare la funzione Lambda

Esegui il processo di operazioni in batch per richiamare la funzione Lambda per la transcodifica dei video. Se il processo non riesce, puoi controllare il report di completamento per identificare la causa.

Per eseguire il processo di operazioni in batch S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Operazioni in batch.
3. Nell'elenco Processi, scegli l'ID processo del processo nella prima riga, ovvero il processo di operazioni in batch S3 creato in precedenza.
4. Scegli Esegui processo.
5. Riesamina i parametri del processo e conferma che il valore di Total objects listed in manifest (Totale oggetti elencati nel manifesto) corrisponda al numero di oggetti nel manifesto. Seleziona quindi Esegui processo.

   Viene visualizzata la pagina del processo di operazioni in batch S3.

6. Dopo l'inizio dell'esecuzione del processo, nella pagina del processo, in Status (Stato) controlla lo stato di avanzamento del processo di operazioni in batch S3, ad esempio Status (Stato), % Complete (% completamento), Total succeeded (rate) (Totale riusciti (tasso)), Total failed (rate) (Totale non riusciti (tasso)), Date terminated (Data di terminazione) e Reason for termination (Motivo della terminazione).

   Al termine del processo di operazioni in batch S3, visualizza i dati nella pagina del processo per confermare che è stato completato come previsto.

   Se oltre il 50% delle operazioni sugli oggetti di un processo di operazione in batch S3 ha esito negativo dopo aver tentato più di 1.000 operazioni, il processo ha automaticamente esito negativo. Per controllare il report di completamento per identificare la causa degli errori, consulta la procedura facoltativa riportata di seguito.

(Facoltativo) Controllo del report di completamento

Puoi utilizzare il report di completamento per determinare quali oggetti non sono riusciti e la causa degli errori.

Come controllare il report di completamento con i dettagli sugli oggetti non eseguiti correttamente

1. Nella pagina del processo di operazioni in batch S3, in Report di completamento, scegli il collegamento per Destinazione del report di completamento.

   Viene aperta la pagina del bucket di destinazione dell'output S3.
2. Nella scheda Oggetti, scegli la cartella che ha il nome che termina con l'ID del processo di operazioni in batch S3 creato in precedenza.
3. Scegli results/ (risultati/).
4. Seleziona la casella di controllo accanto al file `.csv`.
5. Per visualizzare il report del processo, scegli Apri o Scarica.

(Facoltativo) Monitoraggio di ogni chiamata Lambda nella console Lambda

Dopo l'avvio del processo di operazioni in batch S3, il processo richiama la funzione Lambda per ogni oggetto video di input. S3 scrive i log di ogni chiamata Lambda in Logs. CloudWatch Puoi utilizzare il pannello di controllo di monitoraggio della console Lambda per monitorare la funzione Lambda.

1.

Apri la console all'indirizzo https://console.aws.amazon.com/lambda/. AWS Lambda

2. Nel pannello di navigazione a sinistra, scegli Functions (Funzioni).
3. Nell'elenco Funzioni, scegli la funzione Lambda creata nella Fase 4 (ad esempio, **`tutorial-lambda-convert`**).
4. Selezionare la scheda Monitor (Monitora).
5. In Metrics (Parametri), visualizza i parametri di runtime per la funzione Lambda.
6. In Logs, visualizza i dati di log per ogni chiamata CloudWatch Lambda tramite Logs Insights.

> **Note**
>
> Quando utilizzi le operazioni in batch S3 con una funzione Lambda, la funzione Lambda viene richiamata su ciascun oggetto. Se il tuo processo di operazioni in batch S3 è grande, può richiamare più funzioni Lambda contemporaneamente, causando un picco nella simultaneità Lambda.
> Ciascuno Account AWS ha una quota di concorrenza Lambda per regione. Per ulteriori informazioni, consulta Dimensionamento della funzione AWS Lambda nella Guida per gli sviluppatori di AWS Lambda . Una best practice per l'utilizzo delle funzioni Lambda con le operazioni in batch S3 è impostare un limite di simultaneità sulla funzione Lambda stessa. Ciò impedisce al tuo processo di consumare la maggior parte della simultaneità Lambda e potenzialmente di limitare altre funzioni nel tuo account. Per ulteriori informazioni, consulta Gestione della simultaneità riservata Lambda nella Guida per gli sviluppatori di AWS Lambda .

(Facoltativo) Monitora ogni processo di MediaConvert transcodifica video nella console MediaConvert

Un MediaConvert job svolge il lavoro di transcodifica di un file multimediale. Quando il job S3 Batch Operations richiama la funzione Lambda per ogni video, ogni chiamata alla funzione Lambda crea un processo di transcodifica per ogni video in ingresso. MediaConvert

1. Accedi e apri la console all'indirizzo https://console.aws.amazon.com/mediaconvert/. AWS Management Console MediaConvert
2. Se viene visualizzata la pagina MediaConvert introduttiva, scegli Inizia.

3. Nell'elenco Processi, visualizza ogni riga per monitorare l'attività di transcodifica per ogni video di input.
4. Identifica la riga del processo che desideri controllare e scegli il collegamento ID processo per aprire la pagina dei dettagli.
5. Nella pagina Job summary (Riepilogo del processo), in Output, scegli il collegamento per l'output HLS, MP4 o miniature, a seconda di cosa supporta il browser in uso, per passare al bucket S3 di destinazione per i file multimediali di output.
6. Nella cartella corrispondente (HLS, MP4 o Miniature) del bucket S3 di destinazione dell'output, scegli il nome dell'oggetto file multimediale di output.

   Viene aperta la pagina dei dettagli dell'oggetto.
7. Nella pagina dell'oggetto, in Panoramica oggetto, scegli il link in URL oggetto per esaminare il file multimediale di output transcodificato.

Fase 8: Controllo dei file multimediali di output dal bucket S3 di destinazione

Come controllare i file multimediali di output dal bucket S3 di destinazione

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket scegli il nome del bucket S3 di destinazione per i file multimediali di output creati nella Fase 1 (ad esempio, **`tutorial-bucket-2`**).
4. Nella scheda Objecets (Oggetti), ogni video di input ha una cartella con il nome del video di input. Ogni cartella contiene i file multimediali di output transcodificati di un video di input.

   Per controllare i file multimediali di output di un video di input, esegui le seguenti operazioni:

   a. Scegli la cartella con il nome del video di input che desideri controllare.
   b. Scegli la cartella Default/ (Predefinito/).
   c. Scegli la cartella di un formato transcodificato (HLS, MP4 o miniature in questo tutorial).
   d. Scegli il nome del file multimediale di output.
   e. Per esaminare il file transcodificato, nella pagina dell'oggetto scegli il link in URL oggetto.

      I file multimediali di output in formato HLS vengono suddivisi in segmenti brevi. Per riprodurre questi video, incorpora l'URL oggetto del file `.m3u8` in un lettore compatibile.

Fase 9: Pulizia

Se hai transcodificato i video utilizzando S3 Batch Operations, Lambda MediaConvert e solo come esercizio di apprendimento, elimina AWS le risorse che hai allocato in modo da non addebitare più addebiti.

Fasi secondarie

- Eliminazione della configurazione di inventario S3 per il bucket S3 di origine
- Eliminazione della funzione Lambda
- Eliminare il gruppo di CloudWatch log
- Eliminazione dei ruoli IAM e delle policy inline per i ruoli IAM
- Eliminazione della policy IAM gestita dal cliente
- Svuotare i bucket S3
- Eliminazione dei bucket S3

Eliminazione della configurazione di inventario S3 per il bucket S3 di origine

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckets (Bucket) scegli il nome del bucket di origine (ad esempio, **`tutorial-bucket-1`**).
4. Scegliere la scheda Management (Gestione),
5. Nella sezione Configurazioni inventario scegli la configurazione di inventario creata nella Fase 5 (ad esempio, **`tutorial-inventory-config`**).
6. Scegli Elimina e poi Conferma.

Eliminazione della funzione Lambda

1. Apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/.
2. Nel pannello di navigazione a sinistra, scegli Functions (Funzioni).
3. Seleziona la casella di controllo accanto alla funzione creata nella Fase 4 (ad esempio, **`tutorial-lambda-convert`**).

4. Scegli Azioni, quindi Elimina.
5. Nella finestra di dialogo Delete function (Elimina funzione), scegli Delete (Elimina).

Eliminare il gruppo di CloudWatch log

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione a sinistra scegli Log, quindi Gruppi di log.
3. Seleziona la casella di controllo accanto al gruppo di log dal nome che termina con la funzione Lambda creata nella Fase 4 (ad esempio, **`tutorial-lambda-convert`**).
4. Scegli Actions (Operazioni), quindi scegli Delete log group(s) (Elimina gruppi di registri).
5. Nella finestra di dialogo Delete log group(s) (Elimina gruppo/i di log) scegli Delete (Elimina).

Eliminazione dei ruoli IAM e delle policy inline per i ruoli IAM

Per eliminare i ruoli IAM creati nella Fase 2, nella Fase 3, e nella Fase 6 esegui invece le seguenti operazioni:

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegli Ruoli, quindi seleziona la casella di controllo accanto al nome del ruolo che desideri eliminare.
3. Nella parte superiore della pagina, scegli Delete (Elimina).
4. Nella finestra di dialogo di conferma inserisci la risposta richiesta nel campo di inserimento di testo in base al comando e scegli Delete (Elimina).

Eliminazione della policy IAM gestita dal cliente

Per eliminare la policy IAM gestita dal cliente creata nella Fase 6, completa le seguenti operazioni:

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra, seleziona Policy.
3. Scegli il pulsante di opzione accanto alla policy creata nella Fase 6 (ad esempio, **`tutorial-s3batch-policy`**). Puoi utilizzare la casella di ricerca per filtrare l'elenco di policy.

4. Scegli Azioni, quindi Elimina.
5. Conferma di voler eliminare questa policy inserendone il nome nel campo di testo, quindi scegli Elimina.

Svuotare i bucket S3

Per svuotare i bucket S3 creati nei Prerequisiti, nella Fase 1 e nella Fase 5, completa le seguenti operazioni:

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket scegli il pulsante di opzione accanto al nome del bucket che desideri svuotare, quindi scegli Svuota.
4. Nella pagina Svuota bucket conferma che desideri svuotare il bucket inserendo **`permanently delete`** nel campo di testo e quindi scegli Svuota.

Eliminazione dei bucket S3

Per eliminare i bucket S3 creati nei Prerequisiti, nella Fase 1 e nella Fase 5, completa invece le seguenti operazioni:

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket, scegli il pulsante di opzione accanto al nome del bucket che desideri eliminare.
4. Scegli Elimina.
5. Nella pagina Delete bucket (Elimina bucket) conferma che desideri eliminare il bucket inserendone il nome nel campo di testo e quindi scegli Delete bucket (Elimina bucket).

Passaggi successivi

Dopo aver completato questo tutorial, puoi esplorare altri casi d'uso rilevanti:

- Puoi usare Amazon CloudFront per trasmettere i file multimediali transcodificati agli spettatori di tutto il mondo. Per ulteriori informazioni, consulta Tutorial: hosting di video in streaming su richiesta con Amazon S3, Amazon e CloudFront Amazon Route 53.
- Puoi transcodificare i video nel momento in cui li carichi nel bucket di origine S3. A tale scopo, puoi configurare un trigger di eventi Amazon S3 che richiama automaticamente la funzione Lambda con cui transcodificare nuovi oggetti in S3. MediaConvert Per maggiori informazioni consulta Tutorial: Uso di un trigger Amazon S3 per richiamare una funzione Lambda nella Guida per gli sviluppatori di AWS Lambda .

Esercitazione: configurazione di un sito Web statico su Amazon S3

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva negli SDK and. AWS Command Line Interface AWS Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

È possibile configurare un bucket Amazon S3 in modo da funzionare come un sito web. Questo esempio guida attraverso le fasi di hosting di un sito web su Amazon S3.

> **Important**
>
> Il seguente tutorial richiede la disabilitazione dell'opzione Blocco dell'accesso pubblico. È consigliabile mantenere l'impostazione Blocco dell'accesso pubblico abilitata. Se desideri mantenere abilitate tutte e quattro le impostazioni di Block Public Access e ospitare un sito Web statico, puoi utilizzare Amazon CloudFront Origin Access Control (OAC). Amazon CloudFront offre le funzionalità necessarie per configurare un sito Web statico sicuro. I siti Web statici Amazon S3 supportano solo gli endpoint HTTP. Amazon CloudFront utilizza lo storage durevole di Amazon S3 fornendo al contempo intestazioni di sicurezza aggiuntive, come HTTPS. HTTPS aggiunge sicurezza crittografando una normale richiesta HTTP e

proteggendo contro o più comuni attacchi informatici. Per ulteriori informazioni, consulta la sezione Guida introduttiva a un sito Web statico sicuro nella Amazon CloudFront Developer Guide.

Argomenti

Fase 1: creazione di un bucket

Le istruzioni riportate di seguito forniscono una panoramica su come creare i bucket per l'hosting di siti Web. Per step-by-step istruzioni dettagliate sulla creazione di un bucket, consultaCreazione di un bucket.

Per creare un bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Scegliere Create bucket (Crea bucket).
3. Specifica Nome del bucket (ad esempio, **`example.com`**).
4. Scegliere la regione in cui creare il bucket.

   Scegli una regione geografica vicina a te per ridurre al minimo la latenza e i costi e soddisfare i requisiti normativi. La regione scelta determina l'endpoint del sito web Amazon S3. Per ulteriori informazioni, consulta Endpoint del sito Web.
5. Per accettare le impostazioni predefinite e creare il bucket, scegliere Create (Crea).

Fase 2: abilitazione dell'hosting di un sito Web statico

Dopo aver creato un bucket, è possibile abilitare l'hosting di siti Web statici per il bucket. Puoi creare un nuovo bucket o utilizzare un bucket esistente.

Per abilitare l'hosting di un sito Web statico

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Nome bucket, seleziona il nome del bucket per cui desideri abilitare l'hosting di siti Web statici.
3. Scegliere Properties (Proprietà).
4. In Hosting di siti Web statici, seleziona Modifica.
5. Seleziona Utilizza questo bucket per l'hosting di un sito Web.
6. In Hosting di siti Web statici, seleziona Abilita.
7. In Documento di indice immettere il nome file del documento di indice, in genere `index.html`.

   Il nome del documento indice fa distinzione tra maiuscole e minuscole e deve corrispondere esattamente al nome del file del documento indice HTML che si prevede di caricare nel bucket S3. Quando si configura un bucket per l'hosting di siti Web, è necessario specificare un documento di indice. Amazon S3 restituisce questo documento di indice quando si eseguono richieste per il dominio root o per una delle sottocartelle. Per ulteriori informazioni, consulta Configurazione di un documento indice.
8. Per fornire il tuo documento di errore personalizzato per gli errori di classe 4XX, specifica il nome file del documento in Documento di errore.

   Il nome del documento di errore fa distinzione tra maiuscole e minuscole e deve corrispondere esattamente al nome del file del documento di errore HTML che si prevede di caricare nel bucket S3. Se non si specifica un documento di errore personalizzato e si verifica un errore, Amazon S3 restituisce un documento di errore HTML predefinito. Per ulteriori informazioni, consulta Configurazione di un documento di errore personalizzato.
9. (Facoltativo) Per specificare regole di reindirizzamento avanzate, utilizza JSON per descrivere le regole in Regole reindirizzamento.

   Ad esempio, è possibile instradare le richieste in base a prefissi o nomi della chiave dell'oggetto specifici nella richiesta. Per ulteriori informazioni, consulta Configurazione delle regole di reindirizzamento per utilizzare i reindirizzamenti condizionali avanzati.

10. Seleziona Salva modifiche.

    Amazon S3 abilita l'hosting statico del sito web per il tuo bucket. Nella parte inferiore della pagina, in Hosting di siti Web statici, viene visualizzato l'endpoint del sito web per il bucket.

11. In Hosting sito Web statico, prendi nota dell'endpoint.

    Endpoint è l'endpoint del sito web Amazon S3 per il bucket. Dopo aver configurato il bucket come sito Web statico, è possibile utilizzare questo endpoint per testare il sito Web.

Fase 3: modificare le impostazioni di blocco dell'accesso pubblico

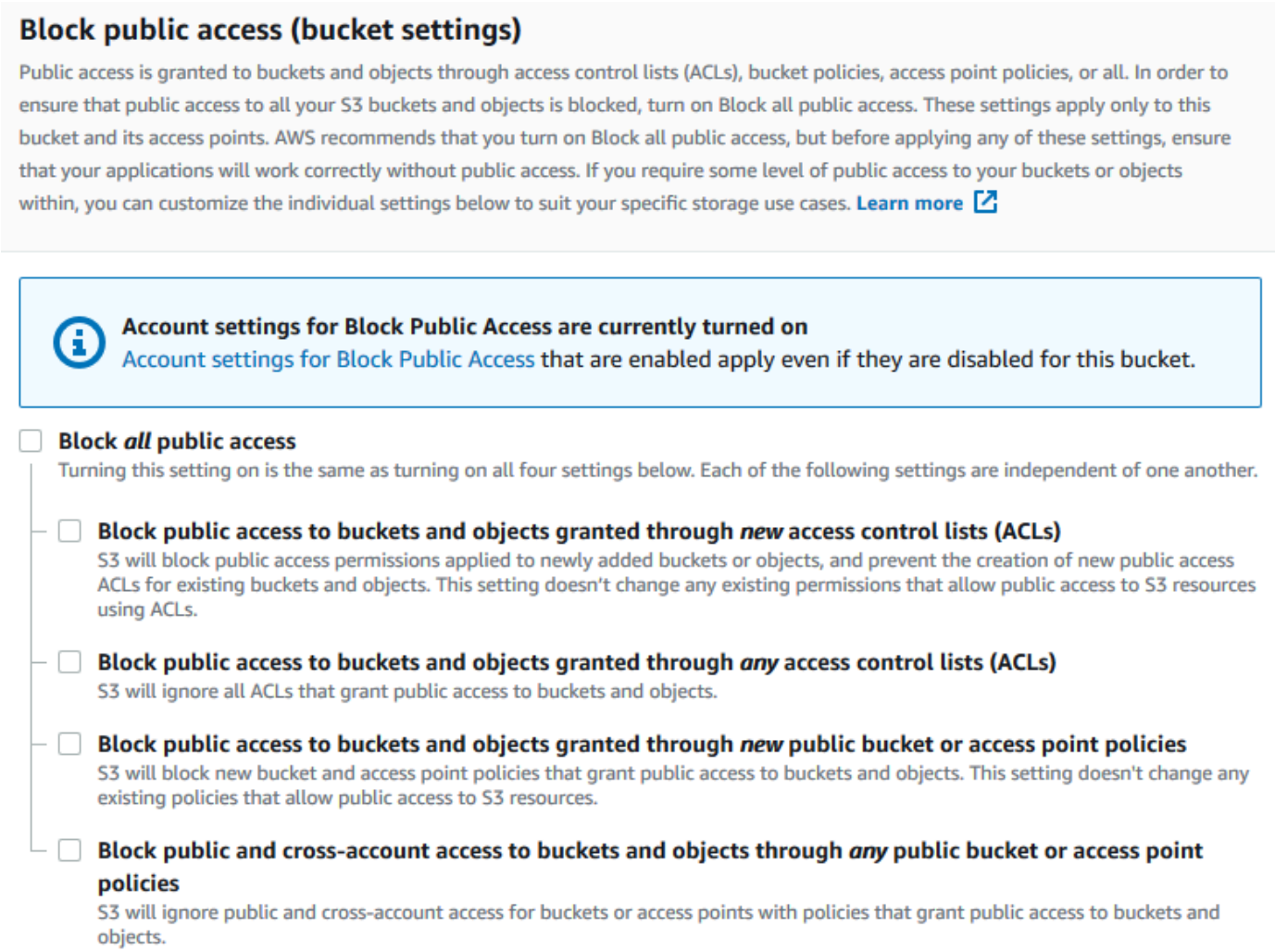
Per impostazione predefinita, Amazon S3 blocca l'accesso pubblico all'account e ai bucket. Per utilizzare un bucket per ospitare un sito Web statico, puoi seguire questa procedura per modificare le impostazioni di blocco dell'accesso pubblico:

> **Warning**
>
> Prima di completare questa fase, consulta Blocco dell'accesso pubblico allo storage Amazon S3 per confermare di avere compreso e accettato i rischi connessi alla concessione di un accesso pubblico. Quando si disattivano le impostazioni di blocco dell'accesso pubblico per rendere pubblico il bucket, chiunque su Internet può accedere al bucket. Consigliamo di bloccare tutti gli accessi pubblici ai bucket.

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
2. Seleziona il nome del bucket configurato come sito Web statico.
3. Seleziona Autorizzazioni.
4. In Blocca accesso pubblico (impostazioni bucket), seleziona Modifica.
5. Deseleziona Blocca tutto l'accesso pubblico, quindi seleziona Salva modifiche.

> **Warning**
>
> Prima di completare questa fase, consulta Blocco dell'accesso pubblico allo storage Amazon S3 per confermare di avere compreso e accettato i rischi connessi alla concessione di un accesso pubblico. Quando si disattivano le impostazioni di blocco

dell'accesso pubblico per rendere pubblico il bucket, chiunque su Internet può accedere al bucket. Consigliamo di bloccare tutti gli accessi pubblici ai bucket.



Amazon S3 disattiva le impostazioni di blocco dell'accesso pubblico per il tuo bucket. Per creare un sito web pubblico statico, potrebbe essere necessario modificare anche le impostazioni di blocco dell'accesso pubblico per l'account prima di aggiungere una policy del bucket. Se le impostazioni dell'account per il blocco dell'accesso pubblico sono attualmente attivate, verrà visualizzata una nota in Blocca accesso pubblico (impostazioni bucket).

Fase 4: aggiunta di una policy del bucket che renda il contenuto del bucket disponibile pubblicamente

Dopo aver modificato le impostazioni di blocco dell'accesso pubblico S3, è possibile aggiungere una policy del bucket per concedere l'accesso pubblico in lettura al bucket. Quando concedi l'accesso pubblico in lettura, chiunque su Internet può accedere al bucket.

> **Important**
>
> La policy seguente è solo un esempio e consente l'accesso completo ai contenuti del bucket. Prima di continuare con questa fase, esamina l'argomento relativo a come proteggere i file nel bucket Amazon S3 per assicurarti di comprendere le best practice per la protezione dei file nel bucket S3 e i rischi connessi alla concessione dell'accesso pubblico .

1. In Bucket, scegli il nome del bucket.
2. Seleziona Autorizzazioni.
3. In Policy del bucket, seleziona Modifica.
4. Per concedere l'accesso in lettura pubblico al sito Web, copiare la policy del bucket seguente e incollarla in Editor della policy del bucket.

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

5. Aggiorna `Resource` al tuo nome bucket.

Nella policy del bucket dell'esempio precedente, *Bucket-Name* è un segnaposto per il nome del bucket. Per utilizzare questa policy di bucket con il proprio bucket, è necessario aggiornare il nome in modo che corrisponda al bucket.

6. Seleziona Salva modifiche.

   Viene visualizzato un messaggio che indica che la policy del bucket è stata aggiunta correttamente.

   Se viene visualizzato l'errore `Policy has invalid resource`, conferma che il nome del bucket nella policy di bucket corrisponde al nome del bucket. Per informazioni sull'aggiunta di una policy del bucket, consulta In che modo aggiungere una policy del bucket S3?

   Se viene visualizzato un messaggio di errore e non è possibile salvare la policy di bucket, controlla le impostazioni di blocco dell'accesso pubblico all'account e al bucket per confermare che consenti l'accesso pubblico al bucket.

Fase 5: configurazione di un documento indice

Quando si abilita l'hosting statico di siti Web per il bucket, si immette il nome del documento di indice (ad esempi, **`index.html`**). Dopo aver abilitato l'hosting di siti Web statici per il bucket, si carica un file HTML con il nome del documento di indice nel bucket.

Per configurare il documento di indice

1. Creare un file `index.html`

   Se non si dispone di un file `index.html`, è possibile utilizzare il seguente codice HTML per crearne 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. Salva il file indice in locale.

   Il nome del file del documento indice deve corrispondere esattamente al nome del documento indice immesso nella finestra di dialogo Hosting sito Web statico. Il nome del documento indice distingue tra maiuscole e minuscole. Ad esempio, se si immette `index.html` per il nome del documento Indice nella finestra di dialogo Hosting sito Web statico, anche il nome del file del documento indice deve essere `index.html` e non `Index.html`.

3. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

4. Nell'elenco S3 buckets (Bucket S3), scegliere il nome del bucket che si desidera utilizzare per ospitare un sito Web statico.

5. Abilitare l'hosting di siti Web statici per il bucket e inserire il nome esatto del documento di indice (ad esempi, `index.html`). Per ulteriori informazioni, consulta Abilitazione dell'hosting di siti Web.

   Dopo aver abilitato l'hosting di siti Web statici, procedere alla fase 6.

6. Per caricare il documento di indice nel bucket, eseguire una delle operazioni seguenti:

   - Trascinare e rilasciare il file di indice nell'elenco bucket della console.
   - Scegliere Upload (Carica) e seguire le istruzioni per scegliere e caricare il file di indice.

   Per step-by-step istruzioni, consultaCaricamento degli oggetti.

7. (Opzionale) Caricare altri contenuti del sito Web nel bucket.

Fase 6: configurare un documento di errore

Quando abiliti l'hosting di siti Web statici per il tuo bucket, specifichi il nome del documento di errore (ad esempio, **`404.html`**). Dopo avere abilitato l'hosting di siti Web statici per il bucket, carichi un file HTML con il nome del documento di errore nel bucket.

Per configurare un documento di errore

1. Crea un documento di errore, ad esempio `404.html`.
2. Salva il file del documento di errore in locale.

Il nome del documento di errore fa distinzione tra maiuscole e minuscole e deve corrispondere esattamente al nome immesso quando hai attivato l'hosting statico di siti Web. Ad esempio, se specifichi `404.html` per il nome del documento di errore nella finestra di dialogo Hosting sito Web statico, anche il nome file del documento di errore dovrà essere `404.html`.

3. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
4. Nell'elenco S3 buckets (Bucket S3), scegliere il nome del bucket che si desidera utilizzare per ospitare un sito Web statico.
5. Abilita l'hosting di siti Web statici per il bucket e inserisci il nome esatto del documento di errore (ad esempio, `404.html`). Per ulteriori informazioni, consulta Abilitazione dell'hosting di siti Web e Configurazione di un documento di errore personalizzato.

   Dopo aver abilitato l'hosting di siti Web statici, procedere alla fase 6.
6. Per caricare il documento di errore nel bucket, completa una delle operazioni riportate di seguito:

   - Trascina e rilascia il file del documento di errore nell'elenco dei bucket della console.
   - Scegliere Upload (Carica) e seguire le istruzioni per scegliere e caricare il file di indice.

   Per step-by-step istruzioni, consultaCaricamento degli oggetti.

Fase 7: testare l'endpoint del sito Web

Dopo aver configurato l'hosting di siti Web statici per il bucket, puoi testare l'endpoint del sito Web.

> **Note**
>
> Amazon S3 non supporta l'accesso HTTPS al sito web. Se desideri utilizzare HTTPS, puoi utilizzare Amazon CloudFront per servire un sito Web statico ospitato su Amazon S3.
> Per ulteriori informazioni, consulta Come si usa CloudFront per servire un sito Web statico ospitato su Amazon S3? e Richiede HTTPS per la comunicazione tra gli spettatori e.
> CloudFront

1. In Bucket, scegli il nome del bucket.
2. Scegliere Properties (Proprietà).

3. Nella parte inferiore della pagina, in Static website hosting (Hosting di siti Web statici), scegliere il proprio Bucket website endpoint (Endpoint del sito web Bucket).

   Il documento indice viene aperto in una finestra del browser separata.

Ora hai un sito Web ospitato su Amazon S3. Questo sito web è disponibile nell'endpoint del sito web Amazon S3. Tuttavia, potresti disporre di un dominio, come `example.com`, che desideri utilizzare per fornire il contenuto dal sito Web creato. Inoltre, potresti voler utilizzare il supporto del dominio root Amazon S3 per servire richieste di `http://www.example.com` e `http://example.com`. Ciò richiede ulteriori fasi. Per un esempio, consulta Tutorial: Configurazione di un sito Web statico utilizzando un dominio personalizzato registrato con Route 53.

Fase 8: Pulizia

Se hai creato il sito Web statico solo come esercizio di apprendimento, elimina le risorse AWS che hai allocato per non accumulare più addebiti. Dopo aver eliminato le AWS risorse, il sito Web non è più disponibile. Per ulteriori informazioni, consulta Eliminazione di un bucket.

Tutorial: Configurazione di un sito Web statico utilizzando un dominio personalizzato registrato con Route 53

Supponiamo che si desideri effettuare l'hosting di un sito Web statico su Amazon S3. Hai registrato un dominio su Amazon Route 53 (ad esempio `example.com`) e desideri che le richieste `http://www.example.com` e `http://example.com` vengano inviate dai tuoi contenuti Amazon S3. È possibile utilizzare questa procedura dettagliata per informazioni su come ospitare un sito web statico e creare reindirizzamenti su Amazon S3 per un sito web con un nome di dominio personalizzato registrato con Route 53. È possibile utilizzare un sito Web esistente che si desira ospitare su Amazon S3 o utilizzare questa procedura dettagliata per iniziare da zero.

Dopo aver completato questa procedura dettagliata, puoi opzionalmente utilizzare Amazon CloudFront per migliorare le prestazioni del tuo sito web. Per ulteriori informazioni, consulta Velocizza il tuo sito Web con Amazon CloudFront.

> **Note**
>
> Gli endpoint del sito Web di Amazon S3 non supportano HTTPS o access point. Se desideri utilizzare HTTPS, puoi utilizzare Amazon CloudFront per servire un sito Web statico ospitato su Amazon S3.

Per un tutorial su come ospitare i tuoi contenuti in modo sicuro con CloudFront Amazon S3, consulta. Tutorial: hosting di video in streaming su richiesta con Amazon S3, Amazon e CloudFront Amazon Route 53 Per ulteriori informazioni, consulta Come si usa CloudFront per servire un sito Web statico ospitato su Amazon S3? e Richiede HTTPS per la comunicazione tra gli spettatori e. CloudFront

Automatizzazione della configurazione di siti Web statici con un modello AWS CloudFormation

È possibile utilizzare un AWS CloudFormation modello per automatizzare la configurazione statica del sito Web. Il AWS CloudFormation modello configura i componenti necessari per ospitare un sito Web statico sicuro in modo che possiate concentrarvi maggiormente sui contenuti del sito Web e meno sulla configurazione dei componenti.

Il AWS CloudFormation modello include i seguenti componenti:

- Amazon S3 – Crea un bucket Amazon S3 per ospitare il tuo sito Web statico.
- CloudFront — Crea una CloudFront distribuzione per velocizzare il tuo sito web statico.
- Lambda@Edge – Utilizza Lambda@Edge per aggiungere intestazioni di sicurezza a ogni risposta del server. Le intestazioni di sicurezza sono un gruppo di intestazioni nella risposta del server Web che indicano ai browser Web di adottare ulteriori precauzioni di sicurezza. Per ulteriori informazioni, consulta il post del blog Aggiungere intestazioni di sicurezza HTTP usando Lambda @Edge e Amazon. CloudFront

Questo AWS CloudFormation modello può essere scaricato e utilizzato. Per informazioni e istruzioni, consulta la sezione Guida introduttiva a un sito Web statico sicuro nella Amazon CloudFront Developer Guide.

Argomenti

- Prima di iniziare
- Fase 1: registrazione di un dominio personalizzato con Route 53
- Fase 2: creare due bucket
- Fase 3: configurazione di un bucket del dominio root per l'hosting di siti Web
- Fase 4: configurare un bucket del sottodominio per il reindirizzamento del sito Web
- Fase 5: configurare la registrazione del traffico del sito Web

Prima di iniziare

Seguendo l'esempio si utilizzeranno i seguenti servizi:

Amazon Route 53 – Usa Route 53 per registrare domini e definire dove instradare il traffico internet per il dominio. L'esempio illustra come creare record di alias Route 53 che instradano il traffico per il dominio (`example.com`) e il sottodominio (`www.example.com`) a un bucket Amazon S3 contenente un file HTML.

Amazon S3 – Viene utilizzato per creare bucket Amazon S3, caricare una pagina di esempio del sito Web, configurare le autorizzazioni per permettere agli utenti di visualizzare il contenuto e configurare i bucket per l'hosting di siti Web.

Fase 1: registrazione di un dominio personalizzato con Route 53

Se non si dispone già di un nome di dominio registrato, ad esempio `example.com`, registrarne uno con Route 53. Per maggiori informazioni, consulta Registrazione di un nuovo dominio nella Guida per gli sviluppatori di Amazon Route 53. Dopo aver registrato il nome di dominio, è possibile creare e configurare i bucket Amazon S3 per l'hosting di siti Web.

Fase 2: creare due bucket

Per le richieste di supporto del dominio root e del sottodominio, occorre creare due bucket:

- Bucket del dominio – `example.com`

- Bucket del sottodominio – `www.example.com`

Questi nomi di bucket devono corrispondere esattamente al nome di dominio. In questo esempio, il nome di dominio è `example.com`. Il contenuto viene ospitato al di fuori del bucket del dominio root (`example.com`). Viene creata una richiesta di reindirizzamento per il bucket del sottodominio (`www.example.com`). In altre parole, se qualcuno accede a `www.example.com` dal proprio browser, viene reindirizzato a `example.com` e visualizza il contenuto ospitato nel bucket Amazon S3 con quel nome.

Per creare i bucket per l'hosting di siti Web

Le istruzioni riportate di seguito forniscono una panoramica su come creare i bucket per l'hosting di siti Web. Per step-by-step istruzioni dettagliate sulla creazione di un bucket, consultaCreazione di un bucket.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Seleziona il bucket del dominio root:
   a. Scegliere Create bucket (Crea bucket).
   b. Specifica Nome del bucket (ad esempio, **`example.com`**).
   c. Scegliere la regione in cui creare il bucket.

      Scegli una regione geografica vicina a te per ridurre al minimo la latenza e i costi e soddisfare i requisiti normativi. La regione scelta determina l'endpoint del sito web Amazon S3. Per ulteriori informazioni, consulta Endpoint del sito Web.
   d. Per accettare le impostazioni predefinite e creare il bucket, scegliere Create (Crea).
3. Crea il tuo bucket del sottodominio:
   a. Scegliere Create bucket (Crea bucket).
   b. Specifica Nome del bucket (ad esempio, **`www.example.com`**).
   c. Scegliere la regione in cui creare il bucket.

      Scegli una regione geografica vicina a te per ridurre al minimo la latenza e i costi e soddisfare i requisiti normativi. La regione scelta determina l'endpoint del sito web Amazon S3. Per ulteriori informazioni, consulta Endpoint del sito Web.
   d. Per accettare le impostazioni predefinite e creare il bucket, scegliere Create (Crea).

Nella fase seguente `example.com` viene configurato per l'hosting di siti Web.

Fase 3: configurazione di un bucket del dominio root per l'hosting di siti Web

In questa fase, configuri il bucket del dominio root (`example.com`) come sito Web. Questo bucket conterrà i contenuti del sito Web. Quando si configura un bucket per l'hosting di siti Web, è possibile accedere al sito web utilizzando il Endpoint del sito Web.

Per abilitare l'hosting di un sito Web statico

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Nome bucket, seleziona il nome del bucket per cui desideri abilitare l'hosting di siti Web statici.
3. Scegliere Properties (Proprietà).
4. In Hosting di siti Web statici, seleziona Modifica.
5. Seleziona Utilizza questo bucket per l'hosting di un sito Web.
6. In Hosting di siti Web statici, seleziona Abilita.
7. In Documento di indice immettere il nome file del documento di indice, in genere `index.html`.

   Il nome del documento indice fa distinzione tra maiuscole e minuscole e deve corrispondere esattamente al nome del file del documento indice HTML che si prevede di caricare nel bucket S3. Quando si configura un bucket per l'hosting di siti Web, è necessario specificare un documento di indice. Amazon S3 restituisce questo documento di indice quando si eseguono richieste per il dominio root o per una delle sottocartelle. Per ulteriori informazioni, consulta Configurazione di un documento indice.
8. Per fornire il tuo documento di errore personalizzato per gli errori di classe 4XX, specifica il nome file del documento in Documento di errore.

   Il nome del documento di errore fa distinzione tra maiuscole e minuscole e deve corrispondere esattamente al nome del file del documento di errore HTML che si prevede di caricare nel bucket S3. Se non si specifica un documento di errore personalizzato e si verifica un errore, Amazon S3 restituisce un documento di errore HTML predefinito. Per ulteriori informazioni, consulta Configurazione di un documento di errore personalizzato.
9. (Facoltativo) Per specificare regole di reindirizzamento avanzate, utilizza JSON per descrivere le regole in Regole reindirizzamento.

Ad esempio, è possibile instradare le richieste in base a prefissi o nomi della chiave dell'oggetto specifici nella richiesta. Per ulteriori informazioni, consulta Configurazione delle regole di reindirizzamento per utilizzare i reindirizzamenti condizionali avanzati.

10. Seleziona Salva modifiche.

    Amazon S3 abilita l'hosting statico del sito web per il tuo bucket. Nella parte inferiore della pagina, in Hosting di siti Web statici, viene visualizzato l'endpoint del sito web per il bucket.

11. In Hosting sito Web statico, prendi nota dell'endpoint.

    Endpoint è l'endpoint del sito web Amazon S3 per il bucket. Dopo aver configurato il bucket come sito Web statico, è possibile utilizzare questo endpoint per testare il sito Web.

Dopo aver modificato le impostazioni di blocco dell'accesso pubblico e aver aggiunto una policy del bucket che consente l'accesso pubblico in lettura, potrai utilizzare l'endpoint del sito Web per accedere al sito Web.

Nella fase successiva viene configurato il sottodominio (`www.example.com`) per reindirizzare le richieste al dominio (`example.com`).

Fase 4: configurare un bucket del sottodominio per il reindirizzamento del sito Web

Una volta che il bucket del dominio root è stato configurato per l'hosting di siti Web, è possibile configurare il bucket del sottodominio per reindirizzare tutte le richieste al dominio. In questo esempio, tutte le richieste per `www.example.com` vengono reindirizzate a `example.com`.

Per configurare una richiesta di reindirizzamento

1. Nella console di Amazon S3, nell'elenco Buckets (Bucket), selezionare il bucket del sottodominio (in questo esempio, `www.example.com`).
2. Scegliere Properties (Proprietà).
3. In Hosting di siti Web statici, seleziona Modifica.
4. Seleziona Reindirizza richieste per un oggetto.
5. Nella casella Target bucket (Bucket di destinazione) immettere il dominio root, ad esempio, **`example.com`**.
6. In Protocol (Protocollo), scegliere HTTP.

7. Seleziona Salva modifiche.

Fase 5: configurare la registrazione del traffico del sito Web

Per tenere traccia del numero di visitatori che accedono al sito Web, puoi abilitare facoltativamente la registrazione per il bucket del dominio root. Per ulteriori informazioni, consulta Registrazione delle richieste con registrazione dell'accesso al server. Se prevedi di utilizzare Amazon CloudFront per velocizzare il tuo sito Web, puoi anche utilizzare CloudFront la registrazione.

Per abilitare la registrazione dell'accesso al server per il bucket del dominio root

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nella stessa regione in cui è stato creato il bucket configurato come sito Web statico, creare un bucket per la registrazione, ad esempio `logs.example.com`.
3. Creare una cartella per i file di registrazione degli accessi al server (ad esempio, `logs`).
4. (Facoltativo) Se desideri utilizzarlo CloudFront per migliorare le prestazioni del tuo sito Web, crea una cartella per i file di CloudFront registro (ad esempio,`cdn`).

   > **Important**
   >
   > Quando crei o aggiorni una distribuzione e abiliti la CloudFront registrazione, CloudFront aggiorna l'elenco di controllo degli accessi ai bucket (ACL) per concedere all'`awslogsdelivery`account le `FULL_CONTROL` autorizzazioni per scrivere i log nel bucket. Per ulteriori informazioni, consulta la sezione Autorizzazioni necessarie per configurare la registrazione standard e accedere ai file di registro nella Amazon CloudFront Developer Guide. Se il bucket che memorizza i log utilizza l'impostazione imposta dal proprietario del bucket per S3 Object Ownership per disabilitare gli ACL, non può scrivere log nel bucket. CloudFront Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

5. Nell'elenco Buckets (Bucket) scegliere il bucket del dominio root.
6. Scegliere Properties (Proprietà).
7. In Registrazione accesso server, seleziona Modifica.
8. Scegli Enable (Abilita).
9. In Bucket di destinazione, seleziona la destinazione del bucket e della cartella per i log di accesso al server:

- Individua la cartella e il percorso del bucket:
  1. Seleziona Sfoglia S3.
  2. Scegli il nome del bucket, quindi seleziona la cartella dei log.
  3. Seleziona Scegli percorso.
- Specifica il percorso del bucket S3, ad esempio, `s3://logs.example.com/logs/`.

10. Seleziona Salva modifiche.

    Nel bucket di log, ora puoi accedere ai tuoi log. Amazon S3 scrive i log di accesso del sito web nel bucket log ogni due ore.

Fase 6: caricare l'indice e il contenuto del sito Web

In questo passaggio carichi il documento di indice e il contenuto facoltativo del sito Web nel bucket del dominio root.

Quando si abilita l'hosting statico di siti Web per il bucket, si immette il nome del documento di indice (ad esempi, **`index.html`**). Dopo aver abilitato l'hosting di siti Web statici per il bucket, si carica un file HTML con il nome del documento di indice nel bucket.

Per configurare il documento di indice

1. Creare un file `index.html`

   Se non si dispone di un file `index.html`, è possibile utilizzare il seguente codice HTML per crearne 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. Salva il file indice in locale.

Il nome del file del documento indice deve corrispondere esattamente al nome del documento indice immesso nella finestra di dialogo Hosting sito Web statico. Il nome del documento indice distingue tra maiuscole e minuscole. Ad esempio, se si immette `index.html` per il nome del documento Indice nella finestra di dialogo Hosting sito Web statico, anche il nome del file del documento indice deve essere `index.html` e non `Index.html`.

3. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
4. Nell'elenco S3 buckets (Bucket S3), scegliere il nome del bucket che si desidera utilizzare per ospitare un sito Web statico.
5. Abilitare l'hosting di siti Web statici per il bucket e inserire il nome esatto del documento di indice (ad esempi, `index.html`). Per ulteriori informazioni, consulta Abilitazione dell'hosting di siti Web.

   Dopo aver abilitato l'hosting di siti Web statici, procedere alla fase 6.
6. Per caricare il documento di indice nel bucket, eseguire una delle operazioni seguenti:

   - Trascinare e rilasciare il file di indice nell'elenco bucket della console.
   - Scegliere Upload (Carica) e seguire le istruzioni per scegliere e caricare il file di indice.

   Per step-by-step istruzioni, consultaCaricamento degli oggetti.
7. (Opzionale) Caricare altri contenuti del sito Web nel bucket.

Fase 7: caricare un documento di errore

Quando abiliti l'hosting di siti Web statici per il tuo bucket, specifichi il nome del documento di errore (ad esempio, **`404.html`**). Dopo avere abilitato l'hosting di siti Web statici per il bucket, carichi un file HTML con il nome del documento di errore nel bucket.

Per configurare un documento di errore

1. Crea un documento di errore, ad esempio `404.html`.
2. Salva il file del documento di errore in locale.

   Il nome del documento di errore fa distinzione tra maiuscole e minuscole e deve corrispondere esattamente al nome immesso quando hai attivato l'hosting statico di siti Web. Ad esempio, se

specifichi `404.html` per il nome del documento di errore nella finestra di dialogo Hosting sito Web statico, anche il nome file del documento di errore dovrà essere `404.html`.

3. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
4. Nell'elenco S3 buckets (Bucket S3), scegliere il nome del bucket che si desidera utilizzare per ospitare un sito Web statico.
5. Abilita l'hosting di siti Web statici per il bucket e inserisci il nome esatto del documento di errore (ad esempio, `404.html`). Per ulteriori informazioni, consulta Abilitazione dell'hosting di siti Web e Configurazione di un documento di errore personalizzato.

   Dopo aver abilitato l'hosting di siti Web statici, procedere alla fase 6.
6. Per caricare il documento di errore nel bucket, completa una delle operazioni riportate di seguito:

   - Trascina e rilascia il file del documento di errore nell'elenco dei bucket della console.
   - Scegliere Upload (Carica) e seguire le istruzioni per scegliere e caricare il file di indice.

   Per step-by-step istruzioni, consultaCaricamento degli oggetti.

Fase 8: modificare le impostazioni dell'accesso pubblico ai blocchi S3

In questo esempio, è possibile modificare le impostazioni di blocco dell'accesso pubblico per il bucket di dominio (`example.com`) per consentire l'accesso pubblico.

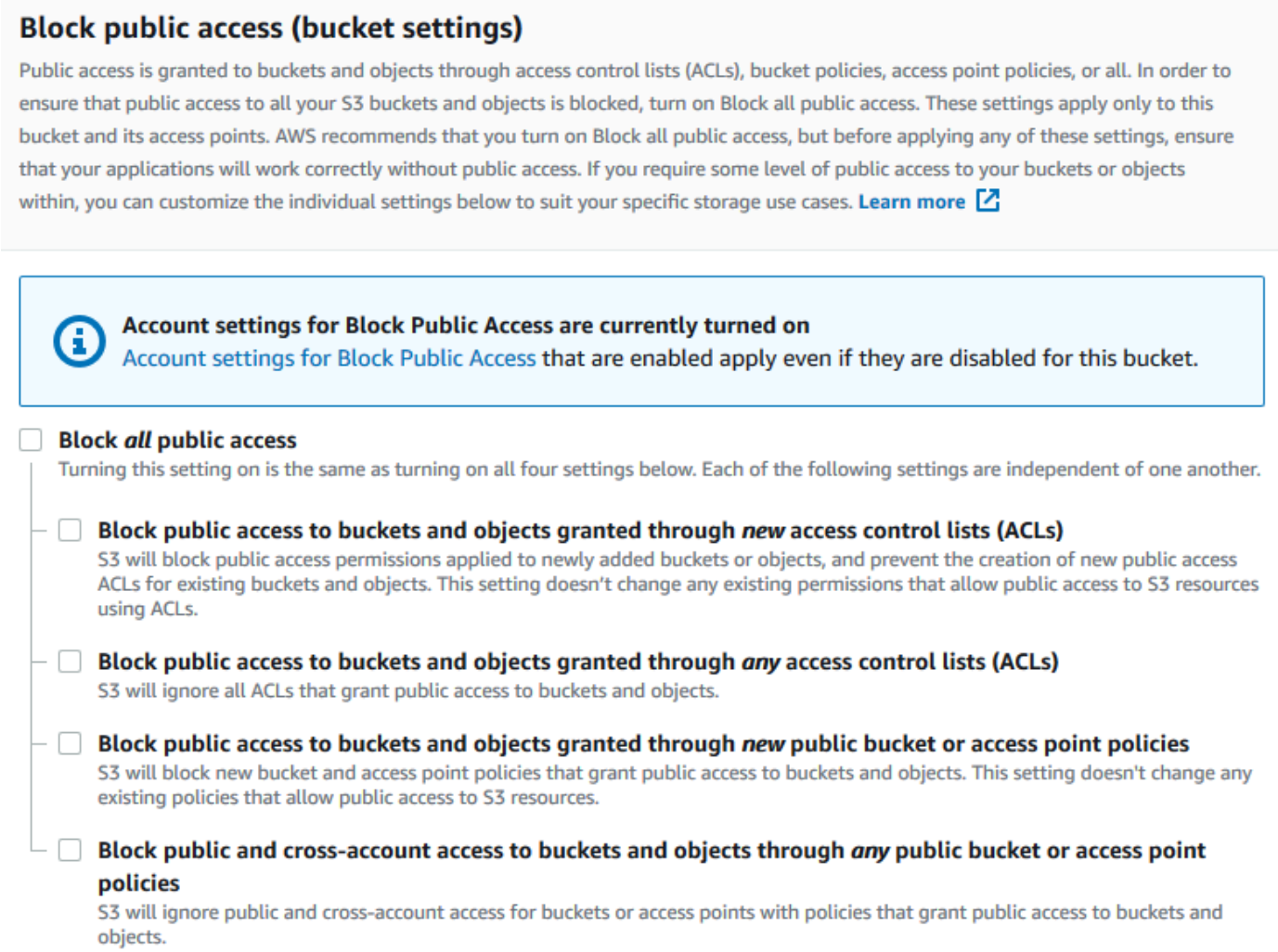
Per impostazione predefinita, Amazon S3 blocca l'accesso pubblico all'account e ai bucket. Per utilizzare un bucket per ospitare un sito Web statico, puoi seguire questa procedura per modificare le impostazioni di blocco dell'accesso pubblico:

> **Warning**
>
> Prima di completare questa fase, consulta Blocco dell'accesso pubblico allo storage Amazon S3 per confermare di avere compreso e accettato i rischi connessi alla concessione di un accesso pubblico. Quando si disattivano le impostazioni di blocco dell'accesso pubblico per rendere pubblico il bucket, chiunque su Internet può accedere al bucket. Consigliamo di bloccare tutti gli accessi pubblici ai bucket.

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.

2. Seleziona il nome del bucket configurato come sito Web statico.
3. Seleziona Autorizzazioni.
4. In Blocca accesso pubblico (impostazioni bucket), seleziona Modifica.
5. Deseleziona Blocca tutto l'accesso pubblico, quindi seleziona Salva modifiche.

> **Warning**
>
> Prima di completare questa fase, consulta Blocco dell'accesso pubblico allo storage Amazon S3 per confermare di avere compreso e accettato i rischi connessi alla concessione di un accesso pubblico. Quando si disattivano le impostazioni di blocco dell'accesso pubblico per rendere pubblico il bucket, chiunque su Internet può accedere al bucket. Consigliamo di bloccare tutti gli accessi pubblici ai bucket.



Amazon S3 disattiva le impostazioni di blocco dell'accesso pubblico per il tuo bucket. Per creare un sito web pubblico statico, potrebbe essere necessario modificare anche le impostazioni di blocco dell'accesso pubblico per l'account prima di aggiungere una policy del bucket. Se le impostazioni dell'account per il blocco dell'accesso pubblico sono attualmente attivate, verrà visualizzata una nota in Blocca accesso pubblico (impostazioni bucket).

Fase 9: collegare una policy del bucket

In questo esempio, si collega una policy del bucket al bucket di dominio (`example.com`) per consentire l'accesso pubblico in lettura. *`Bucket-Name`* nella policy del bucket viene sostituito con il nome del bucket di dominio, ad esempio `example.com`.

Dopo aver modificato le impostazioni di blocco dell'accesso pubblico S3, è possibile aggiungere una policy del bucket per concedere l'accesso pubblico in lettura al bucket. Quando concedi l'accesso pubblico in lettura, chiunque su Internet può accedere al bucket.

> **Important**
>
> La policy seguente è solo un esempio e consente l'accesso completo ai contenuti del bucket. Prima di continuare con questa fase, esamina l'argomento relativo a come proteggere i file nel bucket Amazon S3 per assicurarti di comprendere le best practice per la protezione dei file nel bucket S3 e i rischi connessi alla concessione dell'accesso pubblico .

1. In Bucket, scegli il nome del bucket.
2. Seleziona Autorizzazioni.
3. In Policy del bucket, seleziona Modifica.
4. Per concedere l'accesso in lettura pubblico al sito Web, copiare la policy del bucket seguente e incollarla in Editor della policy del bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
```

```
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::Bucket-Name/*"
            ]
        }
    ]
}
```

5. Aggiorna `Resource` al tuo nome bucket.

   Nella policy del bucket dell'esempio precedente, *`Bucket-Name`* è un segnaposto per il nome del bucket. Per utilizzare questa policy di bucket con il proprio bucket, è necessario aggiornare il nome in modo che corrisponda al bucket.

6. Seleziona Salva modifiche.

   Viene visualizzato un messaggio che indica che la policy del bucket è stata aggiunta correttamente.

   Se viene visualizzato l'errore `Policy has invalid resource`, conferma che il nome del bucket nella policy di bucket corrisponde al nome del bucket. Per informazioni sull'aggiunta di una policy del bucket, consulta In che modo aggiungere una policy del bucket S3?

   Se viene visualizzato un messaggio di errore e non è possibile salvare la policy di bucket, controlla le impostazioni di blocco dell'accesso pubblico all'account e al bucket per confermare che consenti l'accesso pubblico al bucket.

Nella prossima fase, è possibile determinare gli endpoint del sito Web e testare l'endpoint del dominio.

Fase 10: testare l'endpoint del dominio

Dopo aver configurato il bucket di dominio per ospitare un sito Web pubblico, puoi testare l'endpoint. Per ulteriori informazioni, consulta Endpoint del sito Web. Sarai in grado di testare l'endpoint solo per il bucket di dominio, poiché il bucket del sottodominio è impostato per il reindirizzamento del sito Web e non per l'hosting statico del sito Web.

> **Note**
>
> Amazon S3 non supporta l'accesso HTTPS al sito web. Se desideri utilizzare HTTPS, puoi utilizzare Amazon CloudFront per servire un sito Web statico ospitato su Amazon S3.
> Per ulteriori informazioni, consulta Come si usa CloudFront per servire un sito Web statico ospitato su Amazon S3? e Richiede HTTPS per la comunicazione tra gli spettatori e. CloudFront

1. In Bucket, scegli il nome del bucket.
2. Scegliere Properties (Proprietà).
3. Nella parte inferiore della pagina, in Static website hosting (Hosting di siti Web statici), scegliere il proprio Bucket website endpoint (Endpoint del sito web Bucket).

   Il documento indice viene aperto in una finestra del browser separata.

Nella prossima fase ci si servirà di Amazon Route 53 per consentire ai clienti di utilizzare entrambi gli URL personalizzati per spostarsi nel sito.

Fase 11: aggiungere record alias per il dominio e il sottodominio

In questa fase, creare i record alias che si aggiungono alla zona ospitata per le mappe di dominio `example.com` e `www.example.com`. Anziché usare indirizzi IP, i record alias utilizzano gli endpoint dei siti Web Amazon S3. Amazon Route 53 conserva la mappatura dei record alias e degli indirizzi IP dove risiedono i bucket Amazon S3. Creare due record di alias, uno per il dominio root e uno per il sottodominio.

Aggiungere un record di alias per il dominio root e il sottodominio

Per aggiungere un record di alias al dominio root (**`example.com`**)

1. Aprire la console Route 53 all'indirizzo https://console.aws.amazon.com/route53/.

   > **Note**
   >
   > Se non utilizzi già Route 53, consulta la Fase 1: registrare un dominio nella Guida per gli sviluppatori di Amazon Route 53. Dopo la configurazione, puoi tornare alle istruzioni.

2. Scegli Hosted Zones (Zone ospitate).
3. Nell'elenco delle zone ospitate, scegli il none della zona ospitata corrispondente al nome di dominio.
4. Scegli Create record (Crea record).
5. Seleziona Passa alla procedura guidata.

> **Note**
>
> Se desideri utilizzare la creazione rapida per creare i record alias, consulta Configurazione di Route 53 per instradare il traffico a un bucket S3.

6. Scegli Simple routing (Instradamento semplice) e scegli Next (Successivo).
7. Scegli Define simple record (Definisci record semplice).
8. In Record name (Nome del record) accetta il valore predefinito, che è il nome della zona ospitata e del dominio.
9. In Value/Route traffic to (Valore/Instradamento traffico a), seleziona Alias to S3 website endpoint (Alias all'endpoint del sito Web S3).
10. Scegli la regione.
11. Scegli il bucket S3.

    Il nome del bucket deve corrispondere al nome visualizzato nella casella Name (Nome). Nell'elenco Scegli bucket S3, il nome del bucket viene visualizzato con l'endpoint del sito Web di Amazon S3 per la regione in cui è stato creato il bucket, ad esempio, `s3-website-us-west-1.amazonaws.com (example.com)`.

    Scegli bucket S3 riporta un bucket se:

    - Hai configurato il bucket come sito Web statico.
    - Il nome del bucket è uguale al nome del record che stai creando.
    - La corrente Account AWS ha creato il bucket.

    Se il bucket non viene visualizzato nell'elenco Scegli bucket S3, specifica l'endpoint del sito Web di Amazon S3 per la regione in cui è stato creato il bucket, ad esempio **`s3-website-us-west-2.amazonaws.com`**. Per un elenco completo degli endpoint dei siti Web Amazon S3, consulta Endpoint di siti Web Amazon S3. Per maggiori informazioni sulla destinazione alias, consulta Traffico valore/percorso nella Guida per gli sviluppatori di Amazon Route 53.

12. In Tipo di record, scegli A - Indirizza il traffico verso un indirizzo IPv4 e alcune risorse. AWS
13. Per Evaluate target health (Valuta integrità target), seleziona No.
14. Scegli Define simple record (Definisci record semplice).

Per aggiungere un record di alias al sottodominio (**`www.example.com`**)

1. In Configura record, seleziona Definisci record semplice.
2. In Record name (Nome del record) per il sottodominio digita `www`.
3. In Value/Route traffic to (Valore/Instradamento traffico a), seleziona Alias to S3 website endpoint (Alias all'endpoint del sito Web S3).
4. Scegli la regione.
5. Seleziona il bucket S3, ad esempi, `s3-website-us-west-2.amazonaws.com (www.example.com)`.

   Se il bucket non viene visualizzato nell'elenco Scegli bucket S3, specifica l'endpoint del sito Web di Amazon S3 per la regione in cui è stato creato il bucket, ad esempio **`s3-website-us-west-2.amazonaws.com`**. Per un elenco completo degli endpoint dei siti Web Amazon S3, consulta Endpoint di siti Web Amazon S3. Per maggiori informazioni sulla destinazione alias, consulta Traffico valore/percorso nella Guida per gli sviluppatori di Amazon Route 53.
6. In Tipo di record, scegli A - Indirizza il traffico verso un indirizzo IPv4 e alcune risorse. AWS
7. Per Evaluate target health (Valuta integrità target), seleziona No.
8. Scegli Define simple record (Definisci record semplice).
9. Nella pagina Configura record, scegli Crea record.

> **Note**
>
> In genere le modifiche si propagano a tutti i server Route 53 entro 60 secondi. Al termine della propagazione, potrai instradare il traffico al tuo bucket Amazon S3 utilizzando i nomi dei record alias creati in questa procedura.

Aggiungi un record di alias per il dominio root e il sottodominio (vecchia console Route 53)

Per aggiungere un record di alias al dominio root (**`example.com`**)

La console Route 53 è stata riprogettata. Nella console Route 53 è possibile utilizzare temporaneamente la vecchia console. Se scegli di lavorare con la vecchia console Route 53, attenersi alla procedura riportata di seguito.

1. Aprire la console Route 53 all'indirizzo https://console.aws.amazon.com/route53/.

   > **Note**
   >
   > Se non utilizzi già Route 53, consulta la Fase 1: registrare un dominio nella Guida per gli sviluppatori di Amazon Route 53. Dopo la configurazione, puoi tornare alle istruzioni.

2. Scegli Hosted Zones (Zone ospitate).
3. Nell'elenco delle zone ospitate, scegli il none della zona ospitata corrispondente al nome di dominio.
4. Scegliere Create Record Set (Crea set di record).
5. Specifica i seguenti valori:

   Nome

   Accetta il valore predefinito, che è il nome della zona ospitata e del dominio.

   Per il dominio root, non è necessario aggiungere ulteriori informazioni nel campo Name (Nome).

   Tipo

   Selezionare A – IPv4 address (A – indirizzo IPv4).

   Alias

   Scegliere Yes (Sì).

   Destinazione alias

   Nella sezione endpoint del sito Web S3 dell'elenco, scegliere il nome del bucket.

   Il nome del bucket deve corrispondere al nome visualizzato nella casella Name (Nome). Nell'elenco Alias Target (Destinazione alias), il nome bucket è seguito dall'endpoint del sito

web Amazon S3 per la regione in cui è stato creato il bucket, ad esempio `example.com (s3-website-us-west-2.amazonaws.com)`. Alias Target (Destinazione alias) elenca un bucket se:

- Hai configurato il bucket come sito Web statico.
- Il nome del bucket è uguale al nome del record che stai creando.
- La corrente Account AWS ha creato il bucket.

Se il bucket non viene visualizzato nell'elenco di Alias target (Target alias), immetti l'endpoint del sito Web Amazon S3 per la regione in cui è stato creato il bucket, ad esempio `s3-website-us-west-2`. Per un elenco completo degli endpoint dei siti Web Amazon S3, consulta Endpoint di siti Web Amazon S3. Per maggiori informazioni sulla destinazione alias, consulta Traffico valore/percorso nella Guida per gli sviluppatori di Amazon Route 53.

Policy di instradamento

Accettare il valore predefinito Simple (Semplice).

Valutazione dello stato target

Accettare il valore predefinito No.

6. Seleziona Crea.

Per aggiungere un record di alias al sottodominio (**`www.example.com`**)

1. Nella zona ospitata del dominio root (`example.com`), scegliere Create Record Set (Crea set di record).
2. Specifica i seguenti valori:

   Nome

   Per il sottodominio, immettere `www` nella casella.

   Tipo

   Selezionare A – IPv4 address (A – indirizzo IPv4).

   Alias

   Scegliere Yes (Sì).

Destinazione alias

Nella sezione S3 website endpoints (Endpoint del sito web S3) dell'elenco scegliere lo stesso nome di bucket visualizzato nel campo Name (Nome), ad esempio `www.example.com (s3-website-us-west-2.amazonaws.com)`.

Policy di instradamento

Accettare il valore predefinito Simple (Semplice).

Valutazione dello stato target

Accettare il valore predefinito No.

3. Seleziona Crea.

> **Note**
>
> In genere le modifiche si propagano a tutti i server Route 53 entro 60 secondi. Al termine della propagazione, potrai instradare il traffico al tuo bucket Amazon S3 utilizzando i nomi dei record alias creati in questa procedura.

Fase 12: testare il sito Web

Verificare che il sito Web e il reindirizzamento funzionino correttamente. Nel browser, immettere gli URL. In questo esempio, vengono testati i seguenti URL:

- Dominio (`http://example.com`) – Visualizza il documento indice nel bucket `example.com`.
- Sottodominio (`http://www.example.com`) – Reindirizza la richiesta a `http://example.com`. Viene visualizzato il documento di indice nel bucket `example.com`.

Se il tuo sito web o i link di reindirizzamento non funzionano, puoi provare quanto segue:

- Cancella cache – Cancella la cache del tuo browser Web.
- Controlla i server dei nomi – Se la pagina Web e i collegamenti di reindirizzamento non funzionano dopo avere cancellato la cache, puoi confrontare i server dei nomi per il dominio e i server dei nomi per la zona ospitata. Se i server dei nomi non corrispondono, potrebbe essere necessario aggiornare i server dei nomi di dominio in modo che corrispondano a quelli elencati nella zona

ospitata. Per ulteriori informazioni, consulta Aggiunta o modifica di server dei nomi e associazione di record per un dominio.

Dopo aver testato con successo il dominio principale e il sottodominio, puoi configurare una CloudFront distribuzione Amazon per migliorare le prestazioni del tuo sito Web e fornire log da utilizzare per esaminare il traffico del sito Web. Per ulteriori informazioni, consulta Velocizza il tuo sito Web con Amazon CloudFront.

Velocizza il tuo sito Web con Amazon CloudFront

Puoi usare Amazon CloudFront per migliorare le prestazioni del tuo sito Web Amazon S3. CloudFront rende disponibili i file del tuo sito web (come HTML, immagini e video) dai data center di tutto il mondo (note come edge location). Quando un visitatore richiede un file dal tuo sito Web, reindirizza CloudFront automaticamente la richiesta a una copia del file nella edge location più vicina. Ciò determina tempi di download più rapidi rispetto alla richiesta di contenuto da un data center situato più lontano da parte del visitatore.

CloudFront memorizza nella cache i contenuti nelle edge location per un periodo di tempo specificato dall'utente. Se un visitatore richiede contenuti che sono stati memorizzati nella cache per un periodo superiore alla data di scadenza, CloudFront controlla il server di origine per verificare se è disponibile una versione più recente del contenuto. Se è disponibile una versione più recente, CloudFront copia la nuova versione nell'edge location. Le modifiche apportate ai contenuti originali vengono replicate nelle edge location quando i visitatori richiedono i contenuti.

Utilizzo CloudFront senza Route 53

Il tutorial in questa pagina utilizza Route 53 per indicare la tua CloudFront distribuzione. Tuttavia, se desideri fornire contenuti ospitati in un bucket Amazon S3 CloudFront senza utilizzare Route 53, consulta Amazon CloudFront Tutorials: Configurazione di una distribuzione dinamica dei contenuti per Amazon S3. Quando servi contenuti ospitati in un bucket Amazon S3 utilizzando CloudFront, puoi utilizzare qualsiasi nome di bucket e sono supportati sia HTTP che HTTPS.

Configurazione automatica con un modello AWS CloudFormation

Per ulteriori informazioni sull'utilizzo di un AWS CloudFormation modello per configurare un sito Web statico sicuro che crea una CloudFront distribuzione al servizio del tuo sito Web, consulta la sezione Guida introduttiva a un sito Web statico sicuro nella Amazon CloudFront Developer Guide.

Argomenti

- Fase 1: Creare una CloudFront distribuzione
- Passaggio 2: aggiornare il set di record per il dominio e sottodominio
- (Facoltativo) Fase 3: controllare i file di log

Fase 1: Creare una CloudFront distribuzione

Innanzitutto, crei una CloudFront distribuzione. Ciò rende il sito Web accessibile a data center di tutto il mondo.

Per creare una distribuzione con un'origine Amazon S3

1. Apri la CloudFront console all'indirizzohttps://console.aws.amazon.com/cloudfront/v4/home.
2. Scegliere Create Distribution (Crea distribuzione).
3. Nella pagina Create Distribution (Crea distribuzione), nella sezione Origin Settings (Impostazioni origine), digitare l'endpoint del sito Web Amazon S3 per il bucket in Origin Domain Name (Nome dominio di origine), ad esempio **`example.com.s3-website.us-west-1.amazonaws.com`**.

   CloudFront compila l'Origin ID per te.
4. Lasciare i valori predefiniti in Default Cache Behavior Settings (Impostazioni predefinite comportamento cache).

   Con le impostazioni predefinite per Viewer Protocol Policy (Policy protocollo visualizzatore), è possibile utilizzare HTTPS per il sito Web statico. Per ulteriori informazioni su queste opzioni di configurazione, consulta Valori che specifichi quando crei o aggiorni una distribuzione Web nella Amazon CloudFront Developer Guide.
5. In Impostazioni distribuzione, esegui quanto indicato di seguito:

   a. Lascia Classe prezzo impostato su Utilizza tutte le edge location (prestazioni migliori).

   b. Impostare Nomi di dominio alternativi (CNAME) sul dominio root e il sottodominio `www`. In questo tutorial, questi sono rappresentati da `example.com` e `www.example.com`.

   > **Important**
   >
   > Prima di eseguire questa fase, prendi nota dei requisiti per l'utilizzo di nomi di dominio alternativi, in particolare l'esigenza di un certificato SSL/TLS valido.

c. Per SSL Certificate (Certificato SSL), scegliere Custom SSL Certificate (example.com) (Certificato SSL personalizzato (example.com)), quindi scegliere il certificato personalizzato che copre i nomi di dominio e sottodominio.

   Per ulteriori informazioni, consulta il certificato SSL nella Amazon CloudFront Developer Guide.

d. In Default Root Object (Oggetto root predefinito), immettere il nome del documento indice, ad esempio `index.html`.

   Se l'URL utilizzato per accedere alla distribuzione non contiene un nome di file, la CloudFront distribuzione restituisce il documento indice. L'oggetto root predefinito deve corrispondere esattamente al nome del documento indice per il sito Web statico. Per ulteriori informazioni, consulta Configurazione di un documento indice.

e. Imposta Log su On.

   > **Important**
   >
   > Quando crei o aggiorni una distribuzione e abiliti la CloudFront registrazione, CloudFront aggiorna l'elenco di controllo degli accessi ai bucket (ACL) per concedere all'`awslogsdelivery`account le `FULL_CONTROL` autorizzazioni per scrivere i log nel bucket. Per ulteriori informazioni, consulta la sezione Autorizzazioni necessarie per configurare la registrazione standard e accedere ai file di registro nella Amazon CloudFront Developer Guide. Se il bucket che memorizza i log utilizza l'impostazione imposta dal proprietario del bucket per S3 Object Ownership per disabilitare gli ACL, non può scrivere log nel bucket. CloudFront Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

f. In Bucket per log, scegli il bucket log creato.

   Per ulteriori informazioni sulla configurazione di un bucket di registrazione, consulta (Facoltativo) Registrazione del traffico Web.

g. Se desideri archiviare i log generati dal traffico verso la CloudFront distribuzione in una cartella, in Log Prefix, inserisci il nome della cartella.

h. Mantieni i valori predefiniti di tutte le altre impostazioni.

6. Scegliere Create Distribution (Crea distribuzione).

7. Per visualizzare lo stato attuale della distribuzione, cercare la distribuzione nella console e controllare la colonna Status (Stato).

   Lo stato `InProgress` indica che la distribuzione non è ancora completamente distribuita.

   Una volta implementata la distribuzione, puoi fare riferimento ai tuoi contenuti con il nuovo CloudFront nome di dominio.
8. Registra il valore del nome di dominio mostrato nella CloudFront console, `dj4p1rv6mvubz.cloudfront.net` ad esempio.
9. Per verificare che la CloudFront distribuzione funzioni, inserisci il nome di dominio della distribuzione in un browser web.

   Se il tuo sito web è visibile, la CloudFront distribuzione funziona. Se il tuo sito Web ha un dominio personalizzato registrato con Amazon Route 53, avrai bisogno del nome di CloudFront dominio per aggiornare il record impostato nel passaggio successivo.

Passaggio 2: aggiornare il set di record per il dominio e sottodominio

Ora che hai creato correttamente una CloudFront distribuzione, aggiorna il record di alias in Route 53 in modo che punti alla nuova CloudFront distribuzione.

Per aggiornare il record di alias in modo che punti a una distribuzione CloudFront

1. Aprire la console Route 53 all'indirizzo https://console.aws.amazon.com/route53/.
2. Nel riquadro di navigazione a sinistra, scegliere Hosted zones (Zone ospitate).
3. Nella pagina Hosted Zones (Zone ospitate), scegliere la hosted zone creata per il sottodominio, per esempio `www.example.com`.
4. In Records, selezionare il record A creato per il sottodominio.
5. In Record details (Dettagli record), scegliere Edit record (Modifica record).
6. In Indirizza il traffico verso, scegli Alias per CloudFront la distribuzione.
7. In Scegli la distribuzione, scegli la CloudFront distribuzione.
8. Selezionare Salva.
9. Per reindirizzare il record A per il dominio radice alla CloudFront distribuzione, ripeti questa procedura per il dominio radice, `example.com` ad esempio.

   L'aggiornamento ai set di record avviene entro 2-48 ore.

10. Per verificare se i nuovi record A sono effettivi, in un browser Web immetti l'URL del sottodominio, ad esempio `http://www.example.com`.

    Se il browser non reindirizza più al dominio root, ad esempio `http://example.com`, i nuovi record A sono effettivi. Quando il nuovo record A ha effetto, il traffico indirizzato dal nuovo record A alla CloudFront distribuzione non viene reindirizzato al dominio radice. Tutti i visitatori che fanno riferimento al sito utilizzando `http://example.com` o `http://www.example.com` vengono reindirizzati alla CloudFront edge location più vicina, dove possono usufruire di tempi di download più rapidi.

    > **Tip**
    >
    > I browser possono effettuare il caching delle impostazioni di reindirizzamento. Se pensi che le impostazioni del nuovo record A dovrebbero essere diventate effettive ma il tuo browser reindirizza ancora `http://www.example.com` a `http://example.com`, prova a svuotare la cache e a eliminare la cronologia del browser, a chiudere e riaprire la tua applicazione browser o a utilizzare un browser Web differente.

(Facoltativo) Fase 3: controllare i file di log

I log di accesso indicano quante persone stanno visitando il sito Web. Inoltre contengono preziosi dati aziendali che si possono analizzare con altri servizi, come Amazon EMR.

CloudFront i log vengono archiviati nel bucket e nella cartella scelti quando crei una CloudFront distribuzione e abiliti la registrazione. CloudFront scrive i log nel tuo bucket di log entro 24 ore da quando vengono effettuate le richieste corrispondenti.

Per visualizzare i file di log del sito Web

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
2. Scegliere il nome del bucket log per il tuo sito web.
3. Scegli la cartella dei CloudFront log.
4. Scarica i `.gzip` file scritti da CloudFront prima di aprirli.

   Se hai creato il tuo sito Web solo come esercizio di apprendimento, puoi eliminare le risorse che hai allocato per non accumulare più addebiti. A questo proposito, consulta Pulizia delle risorse di esempio. Una volta eliminate le risorse AWS , il sito Web non è più disponibile.

Pulizia delle risorse di esempio

Se il sito Web statico è stato creato come esercizio di apprendimento, elimina le risorse AWS che allocate per non accumulare più addebiti. Una volta eliminate le risorse AWS , il sito Web non è più disponibile.

Attività

- Fase 1: Eliminare la CloudFront distribuzione Amazon
- Passaggio 2: eliminare la zona ospitata Route 53
- Fase 3: disabilitare la registrazione ed eliminare il bucket S3

Fase 1: Eliminare la CloudFront distribuzione Amazon

Prima di eliminare una CloudFront distribuzione Amazon, devi disabilitarla. Una distribuzione disattivata non è più funzionante e non accumula addebiti. Puoi attivare una distribuzione disattivata in qualsiasi momento. Una volta eliminata una distribuzione disattivata, non è più disponibile.

Per disabilitare ed eliminare una CloudFront distribuzione

1. Apri la CloudFront console all'indirizzohttps://console.aws.amazon.com/cloudfront/v4/home.
2. Selezionare la distribuzione che si vuole disattivare e scegliere Disable (Disattiva).
3. Quando viene richiesta la conferma, seleziona Sì, disattiva.
4. Selezionare la distribuzione disattivata e scegliere Delete (Elimina).
5. Quando viene richiesta la conferma, seleziona Sì, elimina.

Passaggio 2: eliminare la zona ospitata Route 53

Prima di eliminare la hosted zone, devi eliminare i set di record creati. Non è necessario eliminare i record NS e SOA; vengono eliminati automaticamente quando elimini la hosted zone.

Per eliminare i set di record

1. Aprire la console Route 53 all'indirizzo https://console.aws.amazon.com/route53/.
2. Nell'elenco dei nomi di dominio, selezionare il nome di dominio, quindi selezionare Vai a set di record.
3. Nell'elenco dei set di record, selezionare le caselle che corrispondono ai record A creati.

Il tipo di ciascun set di record è elencato nella colonna Tipo.

4. Seleziona Elimina set di record.
5. Quando viene richiesta la conferma, seleziona Conferma.

Per eliminare una zona ospitata Route 53

1. Continuando dalla procedura precedente, selezionare Back to Hosted Zones (Torna a zone ospitate).
2. Selezionare il nome di dominio, quindi selezionare Delete Hosted Zone (Elimina hosted zone).
3. Quando viene richiesta la conferma, seleziona Conferma.

Fase 3: disabilitare la registrazione ed eliminare il bucket S3

Prima di eliminare il bucket S3 in uso, accertarsi che le attività di logging siano disattivate per quel bucket. Altrimenti, AWS continua a scrivere i log nel bucket mentre lo elimini.

Per disattivare il log per un bucket

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
2. In Buckets (Bucket), scegliere il nome del bucket e quindi scegliere Properties (Proprietà).
3. Da Properties (Proprietà), selezionare Log.
4. Deseleziona la casella Attivato.
5. Scegliere Save (Salva).

È possibile ora eliminare il bucket. Per ulteriori informazioni, consulta Eliminazione di un bucket.

Creazione, configurazione e utilizzo di bucket Amazon S3

Per memorizzare i tuoi dati in Amazon S3, lavori con risorse denominate bucket e oggetti. Un bucket è un container per oggetti o file. Un oggetto è un file e tutti i metadati che descrivono tale file.

Per memorizzare un oggetto in Amazon S3, crei un bucket e quindi carichi l'oggetto in un bucket. Quando l'oggetto si trova nel bucket, è possibile aprirlo, scaricarlo e spostarlo. Quando non hai più bisogno di un oggetto o di un bucket, puoi ripulire le tue risorse.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

> **Note**
>
> Con Amazon S3 paghi solo per le risorse utilizzate. Per ulteriori informazioni sulle funzionalità e sui prezzi di Amazon S3, consulta Amazon S3. Se sei un nuovo cliente Amazon S3, puoi iniziare a utilizzare Amazon S3 gratuitamente. Per ulteriori informazioni, consulta Piano gratuito AWS.

Gli argomenti di questa sezione forniscono una panoramica dell'utilizzo dei bucket in Amazon S3. Includono informazioni sulla denominazione, la creazione, l'accesso e l'eliminazione di bucket. Per ulteriori informazioni sulla visualizzazione degli oggetti in un bucket, consulta Organizzare, elencare e utilizzare gli oggetti.

Argomenti

- Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3
- Lavorare con Mountpoint per Amazon S3
- Configurazione di trasferimenti veloci e sicuri di file con Amazon S3 Transfer Acceleration
- Utilizzo dei bucket con pagamento a carico del richiedente per utilizzo e trasferimenti di storage
- Restrizioni e limitazioni dei bucket

Panoramica dei bucket

Per caricare i dati (foto, video, documenti ecc.) su Amazon S3, è necessario creare prima un bucket S3 in una delle Regioni AWS.

Un bucket è un container per gli oggetti archiviati in Amazon S3. Puoi archiviare un numero qualsiasi di oggetti in un bucket e avere fino a 100 bucket nel tuo account. Per richiedere un aumento, visita la Console Service Quotas.

Ogni oggetto è contenuto in un bucket. Ad esempio, se l'oggetto denominato `photos/puppy.jpg` è archiviato nel bucket `DOC-EXAMPLE-BUCKET` nella regione Stati Uniti occidentali (Oregon), è indirizzabile tramite l'URL `https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/photos/puppy.jpg`. Per ulteriori informazioni, consulta Accesso a un bucket.

In termini di implementazione, i bucket e gli oggetti sono AWS risorse e Amazon S3 fornisce API per gestirli. Ad esempio, è possibile creare un bucket e caricare oggetti utilizzando l'API di Amazon S3. Per eseguire queste operazioni è possibile utilizzare anche la console di Amazon S3. La console utilizza le API di Amazon S3 per inviare richieste ad Amazon S3.

In questa sezione viene descritto come lavorare con i bucket. Per informazioni sull'utilizzo di oggetti, consulta Panoramica degli oggetti di Amazon S3.

Amazon S3 supporta i bucket globali, il che significa che ogni nome di bucket deve essere univoco all'interno di una partizione. Account AWS Regioni AWS Una partizione è un raggruppamento di regioni. AWS ha attualmente tre partizioni: `aws` (regioni standard), `aws-cn` (regioni Cina) e `aws-us-gov` (AWS GovCloud (US)).

Dopo aver creato un bucket, il nome di quel bucket non può essere utilizzato da un altro Account AWS nella stessa partizione finché il bucket non viene eliminato. Non dovresti dipendere da convenzioni specifiche per la denominazione del bucket per scopi di disponibilità o verifica della sicurezza. Per le linee guida sulla denominazione dei bucket, consulta Regole di denominazione dei bucket.

Amazon S3 crea i bucket nella regione specificata. Per ridurre la latenza, minimizzare i costi o soddisfare i requisiti normativi, scegli Regione AWS quello più vicino a te dal punto di vista geografico. Ad esempio, se risiedi in Europa, potrebbe risultare vantaggiosa la creazione di bucket nella regione Europa (Irlanda) o Europa (Francoforte). Per un elenco delle regioni di Amazon S3, consulta Regioni ed endpoint in Riferimenti generali AWS .

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

> **Note**
>
> Gli oggetti che appartengono a un bucket creato in una regione specifica Regione AWS non lasciano mai quella regione, a meno che non li trasferiate esplicitamente in un'altra regione. Ad esempio, gli oggetti archiviati nella regione Europa (Irlanda) non lasceranno mai tale regione.

Argomenti

Informazioni sulle autorizzazioni

Puoi utilizzare Utente root dell'account AWS le tue credenziali per creare un bucket ed eseguire qualsiasi altra operazione di Amazon S3. Tuttavia, ti consigliamo di non utilizzare le tue credenziali utente root Account AWS per effettuare richieste, ad esempio per creare un bucket. Crea invece un utente AWS Identity and Access Management (IAM) e concedi a quell'utente l'accesso completo (gli utenti per impostazione predefinita non dispongono di autorizzazioni).

Questi utenti sono indicati come amministratori. Puoi utilizzare le credenziali utente amministratore, anziché le credenziali utente root del tuo account, per interagire AWS ed eseguire attività, come creare un bucket, creare utenti e concedere loro autorizzazioni.

Per ulteriori informazioni, consulta Credenziali di Utente root dell'account AWS e credenziali utente IAM nelle sezioni Riferimenti generali AWS e Best practice relative alla sicurezza di IAM della Guida per l'utente di IAM.

Chi crea una risorsa possiede Account AWS quella risorsa. Ad esempio, se crei un utente IAM nel tuo Account AWS e concedi all'utente l'autorizzazione a creare un bucket, l'utente può creare un bucket. Ma l'utente non possiede il bucket; il bucket a Account AWS cui appartiene l'utente possiede il bucket. L'utente necessita di un'autorizzazione aggiuntiva da parte del proprietario delle risorse per eseguire qualsiasi altra operazione sul bucket. Per ulteriori informazioni sulla gestione delle autorizzazioni per le risorse Amazon S3, consulta Identity and Access Management per Amazon S3.

Gestione dell'accesso pubblico ai bucket

L'accesso pubblico viene concesso a bucket e oggetti tramite policy del bucket, liste di controllo accessi (ACL) o entrambi. Amazon S3 offre impostazioni per il blocco dell'accesso pubblico per semplificare la gestione dell'accesso pubblico alle risorse Amazon S3. Le impostazioni di blocco dell'accesso pubblico di Amazon S3 possono sostituire ACL e policy dei bucket in modo da poter imporre limiti uniformi sull'accesso pubblico a queste risorse. Puoi applicare le impostazioni di blocco dell'accesso pubblico a singoli bucket o a tutti i bucket nell'account.

Per garantire che l'accesso pubblico sia bloccato per tutti i bucket e gli oggetti di Amazon S3, tutte e quattro le impostazioni per il Blocco dell'accesso pubblico sono abilitate per default quando si crea un nuovo bucket. Ti consigliamo di attivare tutte e quattro le impostazioni per Blocco dell'accesso pubblico anche per il tuo account. Queste impostazioni bloccano l'accesso pubblico per tutti i bucket correnti e futuri.

Prima di applicare queste impostazioni, verifica che le applicazioni funzionino correttamente senza accesso pubblico. Se è richiesto un certo livello di accesso pubblico ai bucket o agli oggetti, ad esempio per ospitare un sito Web statico come descritto in Hosting di un sito Web statico tramite Amazon S3, puoi personalizzare le impostazioni individuali in funzione dei casi d'uso di storage. Per ulteriori informazioni, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

Tuttavia, è altamente consigliabile mantenere l'impostazione Blocco dell'accesso pubblico abilitata. Se desideri mantenere abilitate tutte e quattro le impostazioni di Block Public Access e ospitare un sito Web statico, puoi utilizzare Amazon CloudFront Origin Access Control (OAC). Amazon CloudFront offre le funzionalità necessarie per configurare un sito Web statico sicuro. I siti Web statici Amazon S3 supportano solo gli endpoint HTTP. Amazon CloudFront utilizza lo storage durevole di Amazon S3 fornendo al contempo intestazioni di sicurezza aggiuntive, come HTTPS. HTTPS

aggiunge sicurezza crittografando una normale richiesta HTTP e proteggendo contro o più comuni attacchi informatici.

Per ulteriori informazioni, consulta la sezione Guida introduttiva a un sito Web statico sicuro nella Amazon CloudFront Developer Guide.

> **Note**
>
> Se viene visualizzato un `Error` quando si elencano i bucket e le relative impostazioni di accesso pubblico, si potrebbe non disporre delle autorizzazioni richieste. Assicurati di disporre delle seguenti autorizzazioni aggiunte alla policy utente o del ruolo:
>
> ```
> s3:GetAccountPublicAccessBlock
> s3:GetBucketPublicAccessBlock
> s3:GetBucketPolicyStatus
> s3:GetBucketLocation
> s3:GetBucketAcl
> s3:ListAccessPoints
> s3:ListAllMyBuckets
> ```
>
> In alcuni rari casi, le richieste possono anche non riuscire a causa di un'interruzione della Regione AWS .

Opzioni di configurazione dei bucket

Amazon S3 supporta varie opzioni di configurazione del bucket. Ad esempio, è possibile configurare il bucket per l'hosting di siti Web, aggiungere una configurazione per la gestione del ciclo di vita degli oggetti nel bucket e configurare il bucket per la registrazione di tutti gli accessi al bucket. Amazon S3 supporta risorse secondarie per l'archiviazione e la gestione delle informazioni di configurazione del bucket. È possibile utilizzare l'API Amazon S3 per creare e gestire queste risorse secondarie. Tuttavia, puoi anche utilizzare la console o gli AWS SDK.

> **Note**
>
> Sono disponibili anche configurazioni a livello di oggetto. Ad esempio, è possibile configurare autorizzazioni a livello di oggetto configurando la lista di controllo accessi (ACL) specifica per quell'oggetto.

In questo caso, si parla di risorse secondarie in quanto esistono nel contesto di un bucket o oggetto specifico. Nella tabella sottostante sono elencate le risorse secondarie che consentono di gestire le configurazioni specifiche per bucket.

Risorsa secondaria	Descrizione
cors (cross-origin resource sharing)	È possibile configurare il bucket per consentire richieste multiorigine. Per ulteriori informazioni, consulta Utilizzo della funzionalità Cross-Origin Resource Sharing (CORS).
event notification	È possibile abilitare il bucket per l'invio di notifiche di specifici eventi del bucket. Per ulteriori informazioni, consulta Notifiche di eventi Amazon S3.
lifecycle	È possibile definire le regole del ciclo di vita per gli oggetti nel bucket che hanno un ciclo di vita ben definito. Ad esempio, è possibile definire una regola per archiviare gli oggetti un anno dopo la creazione o eliminare un oggetto 10 anni dopo la creazione. Per ulteriori informazioni, consulta Gestione del ciclo di vita dello storage.
posizione	Quando crei un bucket, specifichi Regione AWS dove vuoi che Amazon S3 crei il bucket. Amazon S3 archivia queste informazioni nella risorsa secondaria della posizione e fornisce un'API per il recupero di queste informazioni.
logging	La registrazione consente di tenere traccia delle richieste di accesso al bucket. Ogni record del log di accesso contiene i dettagli su una singola richiesta di accesso, ad esempio il richiedente, il nome del bucket, l'ora della richiesta, l'operazione della richiesta, lo stato della risposta e un eventuale codice di errore. Il log di accesso può essere utile nei controlli di accesso e di sicurezza. Può essere utile anche per comprendere la base clienti e la fattura Amazon S3. Per ulteriori informazioni, consulta Registrazione delle richieste con registrazione dell'accesso al server.

Risorsa secondaria	Descrizione
blocco degli oggetti	Per utilizzare il blocco oggetti S3, è necessario abilitarlo per un bucket. Facoltativamente, è anche possibile configurare una modalità e un periodo di conservazione predefiniti che verranno applicati ai nuovi oggetti inseriti nel bucket. Per ulteriori informazioni, consulta Utilizzo del blocco oggetti S3.
policy e ACL (lista di controllo accessi)	Tutte le risorse (come bucket e oggetti) sono private per impostazione predefinita. Amazon S3 supporta opzioni di policy del bucket e liste di controllo accessi (ACL) per la concessione e la gestione delle autorizzazioni a livello di bucket. Amazon S3 archivia le informazioni sulle autorizzazioni nelle risorse secondarie di policy e acl. Per ulteriori informazioni, consulta Identity and Access Management per Amazon S3.
replica	La replica è la copia asincrona e automatica degli oggetti di vari bucket nelle stesse o in diverse Regioni AWS. Per ulteriori informazioni, consulta Panoramica sulla replica degli oggetti.
requestPayment	Per impostazione predefinita, chi crea Account AWS il bucket (il proprietario del bucket) paga per i download dal bucket. Attraverso questa risorsa secondaria, il proprietario del bucket può specificare che il download sarà addebitato alla persona che lo richiede. Amazon S3 fornisce un'API per la gestione di questa risorsa secondaria. Per ulteriori informazioni, consulta Utilizzo dei bucket con pagamento a carico del richiedente per utilizzo e trasferimenti di storage.

Risorsa secondaria	Descrizione
tagging	Puoi aggiungere tag di allocazione dei costi al tuo bucket per classificare e tenere traccia dei costi. AWS Amazon S3 fornisce la risorsa secondaria tagging per archiviare e gestire i tag su un bucket. Utilizzando i tag applicati al bucket, AWS genera un rapporto di allocazione dei costi con utilizzo e costi aggregati in base ai tag. Per ulteriori informazioni, consulta Report di fatturazione e utilizzo per Amazon S3.
transfer acceleration	Transfer Acceleration permette di trasferire i file in modo rapido, semplice e sicuro su lunghe distanze tra il client e un bucket S3. Transfer Acceleration sfrutta le edge location distribuite a livello globale di Amazon CloudFront. Per ulteriori informazioni, consulta Configurazione di trasferimenti veloci e sicuri di file con Amazon S3 Transfer Acceleration.
versioning	La funzione Versioni multiple aiuta a eseguire il ripristino in caso di sovrascritture ed eliminazioni accidentali. Si consiglia l'uso della funzione Versioni multiple come best practice per ripristinare gli oggetti eliminati o sovrascritti per errore. Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.
website	È possibile configurare il bucket per l'hosting di siti Web statici. Amazon S3 archivia questa configurazione creando una risorsa secondaria website. Per ulteriori informazioni, consulta Hosting di un sito Web statico tramite Amazon S3.

Regole di denominazione dei bucket

Per la denominazione dei bucket per uso generico e dei bucket di directory in Amazon S3 si applicano le seguenti regole:

Argomenti

Regole di denominazione dei bucket per uso generico

Per la denominazione dei bucket per uso generico si applicano le seguenti regole.

- I nomi dei bucket devono avere una lunghezza compresa tra 3 (minimo) e 63 (massimo) caratteri.
- I nomi dei bucket possono essere costituiti solo da lettere minuscole, numeri, punti (.) e trattini (-).
- I nomi dei bucket devono iniziare e terminare con una lettera o un numero.
- I nomi dei bucket non devono contenere punti adiacenti.
- I nomi dei bucket non devono avere il formato di un indirizzo IP (ad esempio, 192.168.5.4).
- I nomi dei bucket non devono iniziare con il prefisso `xn--`.
- I nomi dei bucket non devono iniziare con il prefisso `sthree-` o il prefisso. `sthree-configurator`
- I nomi dei bucket non devono terminare con il suffisso`-s3alias`. Questo suffisso è riservato ai nomi alias dei punti di accesso. Per ulteriori informazioni, consulta Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3.
- I nomi dei bucket non devono terminare con il suffisso`--ol-s3`. Questo suffisso è riservato ai nomi alias dei punti di accesso Lambda per oggetti. Per ulteriori informazioni, consulta Come utilizzare un alias in stile bucket per il punto di accesso Lambda per oggetti del bucket S3.
- I nomi dei bucket devono essere univoci Account AWS in tutti gli elementi all'interno di una partizione. Regioni AWS Una partizione è un raggruppamento di regioni. AWS attualmente ha tre partizioni: `aws` (Regioni standard), `aws-cn` (Regioni cinesi) e `aws-us-gov` ().AWS GovCloud (US)
- Il nome di un bucket non può essere utilizzato da un altro Account AWS nella stessa partizione finché il bucket non viene eliminato.
- I bucket utilizzati con Amazon S3 Transfer Acceleration non possono contenere punti (.) nel nome. Per ulteriori informazioni su Transfer Acceleration, consulta Configurazione di trasferimenti veloci e sicuri di file con Amazon S3 Transfer Acceleration.

Per una migliore compatibilità, si consiglia di evitare l'utilizzo di punti (.) nei nomi dei bucket, ad eccezione dei bucket utilizzati solo per l'hosting di siti Web statici. Se includi punti nel nome di un bucket, non puoi utilizzare l' virtual-host-style indirizzamento tramite HTTPS, a meno che non esegui

la convalida del certificato. Questo perché i certificati di sicurezza utilizzati per l'hosting virtuale dei bucket non funzionano per i bucket con punti nei nomi.

Questa limitazione non influisce sui bucket utilizzati per l'hosting di siti Web statici, poiché l'hosting di siti Web statici è disponibile solo su HTTP. Per ulteriori informazioni sull' virtual-host-styleindirizzamento, consulta. Hosting virtuale dei bucket Per ulteriori informazioni sull'hosting di siti Web statici, consulta Hosting di un sito Web statico tramite Amazon S3.

> **Note**
>
> Prima del 1° marzo 2018, i bucket creati nella regione Stati Uniti orientali (Virginia settentrionale) potevano avere nomi lunghi fino a 255 caratteri e con lettere maiuscole e caratteri di sottolineatura. A partire dal 1° marzo 2018, i nuovi bucket nella regione Stati Uniti orientali (Virginia settentrionale) devono essere conformi alle stesse regole applicate in tutte le altre regioni.

Per informazioni sui nomi delle chiavi dell'oggetto, consulta Creazione di nomi di chiavi oggetto.

Esempi di nomi di bucket per uso generico

I seguenti esempi di nomi di bucket sono validi e seguono le linee guida di denominazione consigliate per i bucket per uso generico:

- `docexamplebucket1`
- `log-delivery-march-2020`
- `my-hosted-content`

I seguenti nomi di bucket di esempio sono validi ma non consigliati per usi diversi dall'hosting di siti Web statici:

- `docexamplewebsite.com`
- `www.docexamplewebsite.com`
- `my.example.s3.bucket`

I nomi dei bucket di esempio seguenti non sono validi:

- `doc_example_bucket` (contiene caratteri di sottolineatura)

- `DocExampleBucket` (contiene lettere maiuscole)
- `doc-example-bucket-` (termina con un trattino)

Regole di denominazione dei bucket di directory

I nomi dei bucket di directory devono:

- Sii unico all'interno della zona prescelta Regione AWS e della zona di disponibilità.
- Il nome deve avere una lunghezza compresa tra 3 (min) e 63 (max) caratteri, incluso il suffisso.
- Essere costituiti solo da lettere minuscole, numeri e trattini (-).
- Iniziare e finire con una lettera o un numero.
- Deve includere il seguente suffisso:. `--`*`azid`*`--x-s3`

> **Note**
>
> Quando crei un bucket di directory utilizzando la console, viene aggiunto automaticamente un suffisso al nome di base fornito. Questo suffisso include l'ID zona di disponibilità della zona di disponibilità scelta.
> Quando crei un bucket di directory utilizzando un'API, devi fornire il suffisso completo, incluso l'ID della zona di disponibilità, nella richiesta. Per un elenco degli ID delle zone di disponibilità, consulta. Zone di disponibilità e regioni S3 Express One Zone

Accesso ed elenco di un bucket Amazon S3

Per elencare e accedere ai tuoi bucket Amazon S3, puoi utilizzare diversi strumenti. Rivedi i seguenti strumenti per determinare quale approccio si adatta al tuo caso d'uso:

- Console Amazon S3: con la console Amazon S3, puoi accedere facilmente a un bucket e modificarne le proprietà. Attraverso l'interfaccia utente della console, è possibile eseguire quasi tutte le operazioni sul bucket senza dover scrivere alcun codice.
- AWS CLI: Se devi accedere a più bucket, puoi risparmiare tempo utilizzando il AWS Command Line Interface (AWS CLI) per automatizzare attività comuni e ripetitive. La possibilità di scrivere e di ripetere azioni comuni è una necessità che le aziende considerano frequentemente quando crescono. Per ulteriori informazioni, consulta Sviluppo con Amazon S3 tramite la AWS CLI.

- REST API Amazon S3: la REST API di Amazon S3 ti permette di scrivere programmi e accedere ai bucket in modo programmatico. Amazon S3 supporta un'architettura API in cui i bucket e gli oggetti sono risorse, ciascuna con un URI che identifica in modo univoco la risorsa. Per ulteriori informazioni, consulta Sviluppo con Amazon S3 utilizzando l'API REST.

A seconda del caso d'uso del tuo bucket Amazon S3, esistono diversi metodi consigliati per accedere ai dati sottostanti nei bucket. L'elenco seguente include casi d'uso comuni per l'accesso ai dati.

- Siti Web statici: con Amazon S3 puoi ospitare un sito Web statico. Per questo caso d'uso, è possibile configurare un bucket S3 in modo che funzioni come un sito Web. Per un esempio che illustra le fasi di hosting di un sito Web su Amazon S3, consulta Esercitazione: configurazione di un sito Web statico su Amazon S3.

  Per ospitare un sito Web statico con impostazioni di sicurezza come Block Public Access abilitate, consigliamo di utilizzare Amazon CloudFront con Origin Access Control (OAC) e di implementare intestazioni di sicurezza aggiuntive, come HTTPS. Per ulteriori informazioni, consulta Guida introduttiva a un sito Web statico protetto.

  > **Note**
  >
  > Per l'accesso a un sito Web statico, Amazon S3 supporta sia gli URL in stile hosting virtuale sia quelli in stile percorso. Poiché i bucket sono accessibili tramite URL in stile percorso e URL in stile hosting virtuale, è consigliabile creare bucket con nomi compatibili con DNS. Per ulteriori informazioni, consulta Restrizioni e limitazioni dei bucket.

- Set di dati condivisi: quando si dimensiona su Amazon S3, è comune adottare un modello multi-tenant, in cui si assegnano diversi clienti finali o unità aziendali a prefissi univoci all'interno di un bucket condiviso. Utilizzando i Punti di accesso Amazon S3, puoi suddividere una policy bucket di grandi dimensioni in policy di punti di accesso separate e discrete per ogni applicazione che deve accedere al set di dati condiviso. Questo approccio rende più semplice concentrarsi sulla creazione della giusta policy di accesso per un'applicazione, senza interrompere le attività di tutte le altre applicazioni all'interno del set di dati condiviso. Per ulteriori informazioni, consulta Gestione dell'accesso ai dati con Punti di accesso Amazon S3.
- Carichi di lavoro con velocità di trasmissione effettiva elevata: Mountpoint per Amazon S3 è un client di file open source ad alta velocità di trasmissione effettiva per il montaggio di un bucket Amazon S3 come file system locale. Con Mountpoint, le applicazioni possono accedere agli oggetti archiviati in Amazon S3 tramite operazioni sui file system, quali apertura e lettura. Mountpoint

converte automaticamente queste operazioni in chiamate API a oggetti S3, offrendo alle applicazioni l'accesso all'archiviazione elastica e alla velocità di trasmissione effettiva di Amazon S3 tramite un'interfaccia di file. Per ulteriori informazioni, consulta Lavorare con Mountpoint per Amazon S3.

- Punti di accesso multi-regione: i punti di accesso multi-regione di Amazon S3 forniscono un endpoint globale che le applicazioni possono utilizzare per eseguire le richieste provenienti da bucket S3 situati in Regioni AWS diverse. Puoi utilizzare i punti di accesso multi-regione per creare applicazioni multi-regione con la stessa architettura utilizzata in una singola regione e quindi eseguire tali applicazioni in qualsiasi parte del mondo. Anziché inviare richieste sulla rete Internet pubblica, i punti di accesso multi-regione offrono la resilienza di rete integrata con l'accelerazione delle richieste basate su Internet ad Amazon S3. Per ulteriori informazioni, consulta Punti di accesso multi-regione in Amazon S3.
- Creazione di nuove applicazioni: puoi utilizzare gli AWS SDK per sviluppare applicazioni con Amazon S3. Gli AWS SDK semplificano le attività di programmazione integrando l'API REST di Amazon S3 sottostante. Per creare applicazioni mobili e web connesse, puoi utilizzare gli SDK per AWS dispositivi mobili e la libreria. AWS Amplify JavaScript Per ulteriori informazioni, consulta Sviluppo con Amazon S3 utilizzando gli SDK AWS.
- Secure Shell (SSH) File Transfer Protocol (SFTP): se stai cercando di trasferire in modo sicuro dati sensibili su Internet, puoi utilizzare un server compatibile con SFTP con il tuo bucket Amazon S3. AWS SFTP è un protocollo di rete che supporta tutte le funzionalità di sicurezza e autenticazione di SSH. Questo protocollo ti permette di avere un controllo granulare sull'identità, le autorizzazioni e le chiavi degli utenti, ma se preferisci puoi utilizzare le policy IAM per gestire gli accessi. Per associare un server abilitato per SFTP al tuo bucket Amazon S3, assicurati prima di creare un server abilitato per SFTP. Poi puoi impostare gli account utente e associare il server a un bucket Amazon S3. Per una procedura dettagliata di questo processo, consulta AWS Transfer for SFTP — Servizio SFTP completamente gestito per Amazon S3 nei blog.AWS

Elenco di un bucket

Per elencare tutti i tuoi bucket, devi disporre dell'autorizzazione `s3:ListAllMyBuckets`. Per accedere a un bucket, assicurati di ottenere anche le autorizzazioni AWS Identity and Access Management (IAM) richieste per elencare il contenuto del bucket specificato. Per un esempio di policy di bucket che concede l'accesso a un bucket S3, consulta Concessione a un utente IAM dell'accesso a uno dei bucket. Se si verifica un errore di stato HTTP di accesso negato (403 Forbidden), consulta Policy di bucket e policy IAM.

Puoi elencare il tuo bucket utilizzando la console Amazon S3, o AWS CLI AWS gli SDK.

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Dall'elenco dei bucket per uso generico, scegli il bucket che desideri visualizzare.

> **Note**
>
> L'elenco dei bucket per uso generico include i bucket che si trovano in tutti. Regioni AWS

Usando il AWS CLI

Per utilizzare il AWS CLI per accedere a un bucket S3 o generare un elenco di bucket S3, usa il comando. `ls` Tieni presente che quando elenchi tutti gli oggetti nel bucket, devi disporre dell'autorizzazione `s3:ListBucket`.

Per utilizzare questo comando di esempio, sostituisci *`DOC-EXAMPLE-BUCKET1`* con il nome del tuo bucket.

```
$ aws s3 ls s3://DOC-EXAMPLE-BUCKET1
```

Il seguente comando di esempio elenca tutti i bucket Amazon S3 dell'account:

```
$ aws s3 ls
```

Per ulteriori informazioni e esempi, consulta Elenco di bucket e oggetti.

Utilizzo AWS degli SDK

Puoi anche accedere a un bucket Amazon S3 utilizzando l'operazione API ListBuckets. Per esempi di come utilizzare questa operazione con diversi AWS SDK, consulta. Utilizzo `ListBuckets` con un AWS SDK o una CLI

Creazione di un bucket

Per caricare i dati su Amazon S3, innanzitutto è necessario creare un bucket Amazon S3 in una delle Regioni AWS. Quando si crea un bucket, è necessario scegliere il nome del bucket e una regione. Facoltativamente, è possibile scegliere altre opzioni di gestione dello storage per il bucket. Dopo avere creato un bucket, non è possibile modificare il nome del bucket o la regione. Per ulteriori informazioni sulla denominazione dei bucket, consulta Regole di denominazione dei bucket.

Chi crea Account AWS il bucket ne è proprietario. È possibile caricare un numero qualsiasi di oggetti nel bucket. Per impostazione predefinita, puoi creare fino a 100 bucket in ciascuno dei tuoi. Account AWS Se necessiti di più bucket, puoi aumentare il limite di bucket dell'account fino a un massimo di 1.000 bucket richiedendo un aumento del limite di servizio. Per informazioni su come inviare una richiesta di aumento del limite di bucket, consulta Quote di Servizio AWS in Riferimenti generali di AWS . È possibile archiviare un numero qualsiasi di oggetti in un bucket.

S3 Proprietà dell'oggetto è un'impostazione a livello di bucket Amazon S3 che puoi utilizzare per controllare la proprietà degli oggetti caricati nel bucket e per disabilitare o abilitare le liste di controllo degli accessi (ACL). Per impostazione predefinita, Proprietà dell'oggetto è impostata su Proprietario del bucket applicato e tutte le liste di controllo degli accessi (ACL) sono disabilitate. Con le ACL disabilitate, il proprietario del bucket dispone di ogni oggetto nel bucket e gestisce l'accesso ai dati in maniera esclusiva utilizzando policy.

Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

La crittografia lato server con le chiavi gestite da Amazon S3 (SSE-S3) è il livello di base della crittografia per ogni bucket di Amazon S3. Tutti i nuovi oggetti caricati in un bucket S3 vengono crittografati automaticamente con SSE-S3 come livello base di impostazione della crittografia. Se desideri utilizzare un tipo diverso di crittografia predefinita per la crittografia dei dati, puoi anche specificare la crittografia lato server con chiavi AWS Key Management Service (AWS KMS) o chiavi fornite dal cliente (SSE-C). Per ulteriori informazioni, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3.

Puoi usare la console Amazon S3, le API Amazon S3 AWS o gli SDK per AWS CLI creare un bucket. Per ulteriori informazioni sulle autorizzazioni necessarie per creare un bucket, consulta il riferimento CreateBucketall'API di Amazon Simple Storage Service.

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nella barra di navigazione nella parte superiore della pagina, scegli il nome del file attualmente visualizzato Regione AWS. Quindi, scegli la regione in cui desideri creare un bucket.

   > **Note**
   >
   > Scegli una regione nelle tue vicinanze per ridurre al minimo la latenza e i costi o essere conforme ai requisiti normativi. Gli oggetti archiviati in una regione non la lasciano mai a meno che non vengano trasferiti esplicitamente in un'altra regione. Per un elenco di Amazon S3 Regioni AWS, consulta gli Servizio AWS endpoint in. Riferimenti generali di Amazon Web Services

3. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
4. Scegliere Create bucket (Crea bucket).

   Viene visualizzata la pagina Create bucket (Crea bucket).
5. In Configurazione generale, visualizza Regione AWS dove verrà creato il bucket.
6. In Tipo di bucket, scegli Scopo generale.
7. In Nome bucket, immettere il nome del bucket.

   Il nome del bucket deve:

   - Essere univoco all'interno di una partizione. Una partizione è un raggruppamento di regioni. AWS ha attualmente tre partizioni: `aws` (regioni standard), `aws-cn` (regioni Cina) e `aws-us-gov` (AWS GovCloud (US) Regions).
   - Deve contenere da 3 a 63 caratteri
   - Essere costituito solo da lettere minuscole, numeri, punti (.) e trattini (-). Per una migliore compatibilità, si consiglia di evitare l'utilizzo di punti (.) nei nomi dei bucket, ad eccezione dei bucket utilizzati solo per l'hosting di siti Web statici.
   - Iniziare e finire con una lettera o un numero.

   Una volta creato il bucket, non è possibile modificarne il nome. Per ulteriori informazioni sulla denominazione dei bucket, consulta Regole di denominazione dei bucket.

> **Important**
>
> Evitare di includere informazioni riservate, ad esempio numeri di account, nel nome del bucket. Il nome bucket è visibile nell'URL che punta agli oggetti nel bucket.

8. AWS Management Console ti consente di copiare le impostazioni di un bucket esistente nel tuo nuovo bucket. Se non desideri copiare le impostazioni di un bucket esistente, vai al passaggio successivo.

   > **Note**
   >
   > Questa opzione:
   >
   > - non è disponibile in AWS CLI ed è disponibile solo nella console
   > - Non è disponibile per i bucket di directory
   > - Non copia la policy del bucket dal bucket esistente al nuovo bucket

   Per copiare le impostazioni di un bucket esistente, in Copia le impostazioni dal bucket esistente, seleziona Scegli il bucket. Si apre la finestra Scegli il bucket. Trova il bucket con le impostazioni che desideri copiare e seleziona Scegli il bucket. La finestra Scegli il bucket si chiude e la finestra Crea bucket si riapre.

   In Copia le impostazioni dal bucket esistente, ora vedrai il nome del bucket selezionato. Vedrai anche l'opzione Ripristina i valori predefiniti che puoi usare per rimuovere le impostazioni del bucket copiato. Controlla le impostazioni rimanenti del bucket, nella pagina Crea bucket. Vedrai che ora corrispondono alle impostazioni del bucket che hai selezionato. Puoi passare alla fase finale.

9. Alla voce Proprietà oggetto, per disabilitare o abilitare le ACL e controllare la proprietà degli oggetti caricati nel bucket, scegliere una delle seguenti impostazioni:

   ACL disabilitate

   - Proprietario del bucket applicato (impostazione predefinita): le ACL sono disabilitate e il proprietario del bucket possiede automaticamente e ha il controllo completo di ogni oggetto nel bucket. Le liste di controllo degli accessi (ACL) non influiscono più sulle autorizzazioni di

accesso ai dati nel bucket S3. Il bucket utilizza le policy esclusivamente per definire il controllo degli accessi.

Per impostazione predefinita, le ACL sono disabilitate. La maggior parte degli attuali casi d'uso in Amazon S3 non richiede più l'uso delle ACL. È consigliabile mantenere le ACL disabilitate, tranne in circostanze insolite in cui è necessario controllare individualmente l'accesso per ciascun oggetto. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

ACL abilitate

- Proprietario del bucket scelto – Il proprietario del bucket possiede e ha il pieno controllo sui nuovi oggetti che altri account scrivono nel bucket con l'ACL predefinita `bucket-owner-full-control`.

  Se applichi l'impostazione Proprietario del bucket preferito, per richiedere che tutti i caricamenti di Amazon S3 includano l'ACL predefinita `bucket-owner-full-control`, puoi aggiungere una policy del bucket che consenta solo il caricamento di oggetti che utilizzano questa ACL.
- Scrittore di oggetti: chi carica un oggetto possiede l'oggetto, ne ha il pieno controllo e può concedere ad altri utenti l'accesso ad esso tramite ACL. Account AWS

> **Note**
>
> L'impostazione predefinita è Proprietario del bucket applicato. Per applicare l'impostazione predefinita e mantenere gli ACL disabilitati, è necessaria solo l'autorizzazione `s3:CreateBucket`. Per abilitare gli ACL, è necessario disporre dell'autorizzazione `s3:PutBucketOwnershipControls`.

10. In Impostazioni di blocco dell'accesso pubblico per questo bucket scegli le impostazioni di blocco dell'accesso pubblico che vuoi applicare al bucket.

    Per impostazione predefinita, tutte e quattro le impostazioni Blocco dell'accesso pubblico sono abilitate. È consigliabile mantenere tutte le impostazioni abilitate, a meno che non sia necessario disattivarne una o più di una per il caso d'uso specifico. Per ulteriori informazioni sul blocco dell'accesso pubblico, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

> **Note**
>
> Per abilitare tutte le impostazioni Blocco dell'accesso pubblico, è richiesta solo l'autorizzazione `s3:CreateBucket`. Per disattivare le impostazioni Blocco dell'accesso pubblico, è necessario disporre dell'autorizzazione `s3:PutBucketPublicAccessBlock`.

11. (Facoltativo) In Bucket Versioning (Controllo delle versioni bucket), puoi scegliere se conservare varianti degli oggetti nel bucket. Per ulteriori informazioni sulla funzione Controllo delle versioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

    Per disabilitare o abilitare il controllo delle versioni nel bucket, scegli Disable (Disabilita) o Enable (Abilita).

12. (Facoltativo) In Tags (Tag), puoi scegliere di aggiungere tag al bucket. I tag sono coppie chiave-valore utilizzate per classificare lo spazio di archiviazione.

    Per aggiungere un tag al bucket, inserisci un valore in Key (Chiave) e facoltativamente un valore in Value (Valore), quindi scegli Add Tag (Aggiungi tag).

13. In Default encryption (Crittografia di default), scegliere Edit (Modifica).

14. Per configurare la crittografia predefinita, in Tipo di crittografia scegli una delle seguenti opzioni:

    - Chiavi gestite Amazon S3 (SSE-S3)
    - AWS Key Management Service chiave (SSE-KMS)

    > **Important**
    >
    > Se usi l'opzione SSE-KMS per la configurazione della crittografia predefinita, sei soggetto alla quota delle richieste al secondo di AWS KMS. Per ulteriori informazioni sulle AWS KMS quote e su come richiedere un aumento delle quote, consulta Quotas nella Developer Guide.AWS Key Management Service

    I bucket e i nuovi oggetti sono crittografati con la crittografia lato server con una chiave gestita da Amazon S3 come livello base di configurazione della crittografia. Per ulteriori informazioni sulla crittografia predefinita, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3.

Per ulteriori informazioni sull'utilizzo della crittografia lato server di Amazon S3 per crittografare i dati, consulta Uso della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).

15. Se scegli Chiave AWS Key Management Service (SSE-KMS), procedi come segue:

    a. In Chiave AWS KMS specifica la tua chiave KMS in uno dei seguenti modi:

    - Per scegliere da un elenco di chiavi KMS disponibili, scegli tra le tue AWS KMS keys chiavi KMS e scegli la tua chiave KMS dall'elenco delle chiavi disponibili.

      In questo elenco vengono visualizzate sia la chiave Chiave gestita da AWS (`aws/s3`) che quella gestita dai clienti. Per ulteriori informazioni sulle chiavi gestite dal cliente, consulta Chiavi gestite dal cliente e chiavi AWS nella Guida per gli sviluppatori di AWS Key Management Service .

    - Per specificare l'ARN della chiave KMS, scegli Inserisci l'ARN della AWS KMS key e quindi specifica l'ARN della chiave KMS nel campo visualizzato.

    - Per creare una nuova chiave gestita dal cliente nella AWS KMS console, scegli Crea una chiave KMS.

      Per ulteriori informazioni sulla creazione di una AWS KMS key, consulta Creating keys nella AWS Key Management Service Developer Guide.

      Important

      Puoi utilizzare solo le chiavi KMS disponibili nello Regione AWS stesso bucket. La console Amazon S3 elenca solo le prime 100 chiavi KMS nella stessa regione del bucket. Per utilizzare una chiave KMS non elencata, devi inserire l'ARN della chiave KMS. Se desideri utilizzare una chiave KMS di proprietà di un account diverso, è necessario innanzitutto disporre dell'autorizzazione necessaria per l'uso della chiave e quindi inserire l'ARN della chiave KMS. Per ulteriori informazioni sulle autorizzazioni tra account per le chiavi KMS, consulta Creazione di chiavi KMS utilizzabili da altri account nella Guida per gli sviluppatori di AWS Key Management Service . Per ulteriori informazioni su SSE-KMS, consulta Specifica della crittografia lato server con AWS KMS (SSE-KMS).
      Quando utilizzi una chiave KMS AWS KMS key per la crittografia lato server in Amazon S3, devi scegliere una chiave KMS di crittografia simmetrica. Amazon S3 supporta solo chiavi KMS di crittografia simmetriche e non chiavi KMS asimmetriche.

Per ulteriori informazioni, consulta Identificazione delle chiavi KMS simmetriche e asimmetriche nella Guida per gli sviluppatori di AWS Key Management Service .

Per ulteriori informazioni sulla creazione di una AWS KMS key, consulta Creating keys nella Developer Guide.AWS Key Management Service Per ulteriori informazioni sull'utilizzo AWS KMS con Amazon S3, consulta. Utilizzo della crittografia lato server con chiavi (SSE-KMS) AWS KMS

b. Quando configuri il bucket per utilizzare la crittografia predefinita con SSE-KMS puoi anche abilitare le chiavi bucket S3. S3 Bucket Keys riduce il costo della crittografia diminuendo il traffico di richieste da Amazon S3 a. AWS KMS Per ulteriori informazioni, consulta Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3.

   Per utilizzare le chiavi bucket S3, in Chiave bucket seleziona Abilita.

16. (Facoltativo) Se si desidera abilitare il blocco oggetti S3, effettua le seguenti operazioni:

    a. Scegli Impostazioni avanzate.

    > **Important**
    >
    > L'abilitazione del blocco oggetti consente anche la funzione Controllo delle versioni del bucket. Dopo averlo abilitato, per il blocco di oggetti è necessario configurare le impostazioni predefinite di conservazione e di blocco di carattere legale per proteggere i nuovi oggetti dall'eliminazione o dalla sovrascrittura.

    b. Se desideri abilitare il blocco degli oggetti, scegli Enable (Abilita), leggi l'avviso visualizzato e confermalo.

    Per ulteriori informazioni, consulta Utilizzo del blocco oggetti S3.

    > **Note**
    >
    > Per creare un bucket abilitato per il blocco degli oggetti, devi disporre delle seguenti autorizzazioni: `s3:CreateBucket`, `s3:PutBucketVersioning` e `s3:PutBucketObjectLockConfiguration`.

17. Seleziona Crea bucket.

AWS Utilizzo degli SDK

Quando utilizzi gli AWS SDK per creare un bucket, devi creare un client e quindi utilizzare il client per inviare una richiesta di creazione di un bucket. Come best practice, crea il client e il bucket nella stessa Regione AWS. Se non specifichi una regione quando crei un client o un bucket, Amazon S3 utilizza la regione predefinita Stati Uniti orientali (Virginia settentrionale). Se vuoi limitare la creazione del bucket a una Regione AWS specifica, utilizza la chiave di condizione `LocationConstraint`.

Per creare un client per accedere a un endpoint dual-stack, è necessario specificare una Regione AWS. Per ulteriori informazioni, consulta Endpoint dual-stack. Per un elenco di quelli disponibili Regioni AWS, consulta Regioni ed endpoint in. Riferimenti generali di AWS

Quando si crea un client, la regione viene mappata all'endpoint specifico della regione. Il client utilizza questo endpoint per comunicare con Amazon S3: `s3.region.amazonaws.com`. Se la tua regione è stata lanciata dopo il 20 marzo 2019, il tuo client e il tuo bucket devono trovarsi nella stessa regione. Puoi comunque utilizzare un client nella regione Stati Uniti orientali (Virginia settentrionale) per creare un bucket in qualsiasi regione lanciata prima del 20 marzo 2019. Per ulteriori informazioni, consulta Endpoint legacy.

Questi esempi di codice AWS SDK eseguono le seguenti attività:

- Creare un client specificando esplicitamente una Regione AWS: nell'esempio, il client utilizza l'endpoint `s3.us-west-2.amazonaws.com` per comunicare con Amazon S3. Puoi specificare qualsiasi Regione AWS. Per un elenco di Regioni AWS, consulta Regioni ed endpoint nel Riferimento AWS generale.
- Inviare una richiesta di creazione di bucket specificando solo il nome del bucket: il client invia ad Amazon S3 la richiesta di creare il bucket nella regione in cui hai creato un client.
- Recuperare le informazioni sulla posizione del bucket: Amazon S3 memorizza le informazioni sulla posizione del bucket nella risorsa secondaria posizione associata al bucket.

Java

Questo esempio mostra come creare un bucket Amazon S3 utilizzando la AWS SDK for Java. Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
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

Per informazioni su come creare e testare un esempio funzionante, consulta AWS SDK for .NET Version 3 API Reference.

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

Per informazioni su come creare e testare un esempio funzionante, vedete AWS SDK for Ruby - Versione 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__
```

Usando il AWS CLI

Puoi anche usare AWS Command Line Interface (AWS CLI) per creare un bucket S3. Per ulteriori informazioni, consulta create-bucket nella Guida di riferimento ai comandi della AWS CLI .

Per informazioni su AWS CLI, consulta What is the? AWS Command Line Interface nella Guida AWS Command Line Interface per l'utente.

Visualizzazione delle proprietà di un bucket S3

Puoi visualizzare le proprietà di qualsiasi bucket Amazon S3 di tua proprietà. Queste impostazioni includono quanto segue:

- Funzione versioni multiple bucket: consente di gestire più versioni di un oggetto in un unico bucket utilizzando la funzione Versioni multiple. Per default, la funzione Versioni multiple è disabilitata per un nuovo bucket. Per informazioni sull'abilitazione della funzione Versioni multiple, consulta Abilitazione della funzione Controllo delle versioni sui bucket.
- Tag: con l'allocazione AWS dei costi, puoi utilizzare i bucket tag per annotare la fatturazione relativa all'utilizzo di un bucket. Un tag è una coppia chiave-valore che rappresenta un'etichetta assegnata a un bucket. Per ulteriori informazioni, consulta Utilizzo dei tag per l'allocazione dei costi per i bucket S3.
- Crittografia predefinita: l'abilitazione della crittografia predefinita fornisce la crittografia automatica lato server. Amazon S3 crittografa di un oggetto prima di salvarlo su disco e lo decritta quando lo scarichi. Per ulteriori informazioni, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3.
- Registrazione accessi al server: fornisce record dettagliati per le richieste effettuate al bucket con la registrazione degli accessi al server. Per default, Amazon S3 non raccoglie i log degli accessi al server. Per informazioni sull'attivazione della registrazione degli accessi al server, consulta. Abilitazione della registrazione degli accessi al server Amazon S3
- AWS CloudTrail eventi relativi ai dati: CloudTrail da utilizzare per registrare gli eventi relativi ai dati. Per impostazione predefinita, i trail non registrano gli eventi di dati Per gli eventi di dati sono previsti costi aggiuntivi. Per ulteriori informazioni, consulta Registrazione di eventi di dati per i trail nella Guida per l'utente di AWS CloudTrail .

- Notifiche eventi: è possibile fare in modo che al verificarsi di determinati eventi di bucket Amazon S3 vengano inviati messaggi di notifica a una destinazione. Per ulteriori informazioni, consulta Attivazione e configurazione delle notifiche di eventi tramite la console di Amazon S3.
- Accelerazione trasferimento: permette di trasferire i file in modo rapido, semplice e sicuro su lunghe distanze tra il client e un bucket S3. Per informazioni sull'abilitazione dell'accelerazione di trasferimento, consulta Abilitazione e utilizzo di S3 Transfer Acceleration.
- Blocco oggetto: utilizza S3 Object Lock per impedire che un oggetto venga eliminato o sovrascritto per un periodo di tempo fisso o indefinito. Per ulteriori informazioni, consulta Utilizzo del blocco oggetti S3.
- Pagamento a carico del richiedente: questa funzione fa in modo che il pagamento delle richieste e dei trasferimenti dei dati sia a carico del richiedente anziché del proprietario del bucket. Per ulteriori informazioni, consulta Utilizzo dei bucket con pagamento a carico del richiedente per utilizzo e trasferimenti di storage.
- Hosting sito Web statico: puoi ospitare un sito Web statico su Amazon S3. Per ulteriori informazioni, consulta Hosting di un sito Web statico tramite Amazon S3.

Puoi visualizzare le proprietà del bucket utilizzando AWS Management Console AWS CLI, o SDK AWS

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets seleziona il nome del bucket del quale desideri visualizzare le proprietà.
3. Scegliere la scheda Properties (Proprietà).
4. Nella pagina Proprietà, puoi configurare le proprietà precedenti per il bucket.

Usando il AWS CLI

Visualizza le proprietà del bucket con AWS CLI

I comandi seguenti mostrano come utilizzare il AWS CLI per elencare diverse proprietà del bucket.

Quanto segue restituisce il set di tag associato al bucket example-s3-bucket1. Per ulteriori informazioni sui bucket tag, consulta,. Utilizzo dei tag per l'allocazione dei costi per i bucket S3

```
aws s3api get-bucket-tagging --bucket example-s3-bucket1
```

Per ulteriori informazioni ed esempi, consulta get-bucket-tagging nel Riferimento ai comandi AWS CLI .

Quanto segue restituisce lo stato di controllo delle versioni del bucket example-s3-bucket1. Per informazioni sul controllo delle versioni del bucket, consulta. Utilizzo della funzione Controllo delle versioni nei bucket S3

```
aws s3api get-bucket-versioning --bucket example-s3-bucket1
```

Per ulteriori informazioni ed esempi, consulta get-bucket-versioning nel Riferimento ai comandi AWS CLI .

Quanto segue restituisce la configurazione di crittografia predefinita per il bucket example-s3-bucket1. Per impostazione predefinita, tutti i bucket hanno una configurazione di crittografia predefinita che utilizza la crittografia lato server con chiavi gestite di Amazon S3 (SSE-S3). Per informazioni sulla crittografia predefinita del bucket, consulta. Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3

```
aws s3api get-bucket-encryption --bucket example-s3-bucket1
```

Per ulteriori informazioni ed esempi, consulta get-bucket-encryption nel Riferimento ai comandi AWS CLI .

Quanto segue restituisce la configurazione di notifica del bucket example-s3-bucket1. Per informazioni sulle notifiche degli eventi del bucket, consulta. Notifiche di eventi Amazon S3

```
aws s3api get-bucket-notification-configuration --bucket example-s3-bucket1
```

Per ulteriori informazioni ed esempi, consulta get-bucket-notification-configuration nel Riferimento ai comandi AWS CLI .

Quanto segue restituisce lo stato di registrazione per il bucket example-s3-bucket1. Per informazioni sulla registrazione del bucket, vedere. Registrazione delle richieste con registrazione dell'accesso al server

```
aws s3api get-bucket-logging --bucket example-s3-bucket1
```

Per ulteriori informazioni ed esempi, consulta get-bucket-logging nel Riferimento ai comandi AWS CLI .

Utilizzo degli SDK AWS

Per esempi su come restituire le proprietà dei bucket con gli AWS SDK, come il controllo delle versioni, i tag e altro, consulta. Azioni per Amazon S3 tramite SDK AWS

Per informazioni generali sull'utilizzo di diversi AWS SDK, consulta. Sviluppo con Amazon S3 utilizzando gli SDK AWS

Svuotamento di un bucket

Puoi svuotare il contenuto di un bucket utilizzando la console Amazon S3 AWS , gli SDK o (). AWS Command Line Interface AWS CLI Quando si svuota un bucket, si elimina tutto il suo contenuto, ma si mantiene il bucket. Lo svuotamento di un bucket non è reversibile. Anche gli oggetti aggiunti al bucket mentre l'operazione di svuotamento del bucket è in corso potrebbero essere eliminati. Tutti gli oggetti (incluse tutte le versioni degli oggetti e i marker di eliminazione) nel bucket devono essere eliminati prima che possa essere eliminato il bucket stesso.

Quando si svuota un bucket che ha il controllo delle versioni S3 abilitato o sospeso, tutte le versioni di tutti gli oggetti nel bucket vengono eliminate. Per ulteriori informazioni, consulta Utilizzo di oggetti in un bucket che supporta la funzione Controllo delle versioni.

Inoltre è possibile specificare una configurazione del ciclo di vita su un bucket per predisporre la scadenza degli oggetti in modo che Amazon S3 li possa eliminare. Per ulteriori informazioni, consulta Impostazione di una configurazione del ciclo di vita su un bucket. Per svuotare un bucket di grandi dimensioni, ti consigliamo di utilizzare una regola di configurazione del ciclo di vita S3. La scadenza del ciclo di vita è un processo asincrono, pertanto l'esecuzione della regola potrebbe richiedere alcuni giorni prima che il bucket sia vuoto. Dopo la prima volta che Amazon S3 esegue la regola, tutti gli oggetti idonei alla scadenza vengono contrassegnati per l'eliminazione. Non vengono più addebitati costi per gli oggetti contrassegnati per l'eliminazione. Per ulteriori informazioni, consulta Come posso svuotare un bucket Amazon S3 utilizzando una regola di configurazione del ciclo di vita?.

Utilizzo della console S3

È possibile utilizzare la console Amazon S3 per svuotare un bucket, ossia eliminare tutti gli oggetti nel bucket senza eliminare il bucket.

Per svuotare un bucket S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Nome del bucket, scegli l'opzione accanto al nome del bucket che desideri svuotare, quindi seleziona Svuota.
3. Nella pagina Empty bucket (Svuota bucket) confermare che si desidera svuotare il bucket immettendo il nome del bucket nel campo di testo e quindi scegliere Empty (Svuota).
4. Monitorare l'avanzamento del processo di svuotamento del bucket nella pagina Svuota bucket: stato.

Usando il AWS CLI

È possibile svuotare un bucket utilizzando il AWS CLI solo se il bucket non ha il Bucket Versioning abilitato. Se il controllo delle versioni non è abilitato, puoi utilizzare il AWS CLI comando `rm` (remove) con il `--recursive` parametro per svuotare il bucket (o rimuovere un sottoinsieme di oggetti con un prefisso specifico per il nome della chiave).

Il comando `rm` rimuove gli oggetti con prefisso del nome della chiave `doc`, ad esempio `doc/doc1` e `doc/doc2`.

```
$ aws s3 rm s3://bucket-name/doc --recursive
```

Per rimuovere tutti gli oggetti senza specificare un prefisso, è necessario utilizzare il comando seguente.

```
$ aws s3 rm s3://bucket-name --recursive
```

Per ulteriori informazioni, consulta Utilizzo dei comandi di alto livello S3 con la AWS CLI nella Guida per l'utente di AWS Command Line Interface .

> **Note**
>
> Non è possibile rimuovere oggetti da un bucket su cui è abilitata la funzione Versioni multiple. Con questo comando, Amazon S3 aggiunge un contrassegno di eliminazione quando elimini un oggetto. Per ulteriori informazioni sulla funzione Versioni multiple del bucket S3, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Utilizzo degli SDK AWS

Puoi utilizzare gli AWS SDK per svuotare un bucket o rimuovere un sottoinsieme di oggetti con un prefisso di nome chiave specifico.

Per un esempio di come svuotare un bucket utilizzando, vedi. AWS SDK for JavaEliminazione di un bucket Il codice elimina tutti gli oggetti, indipendentemente dal fatto che sul bucket sia abilitata la funzione Versioni multiple o meno, quindi elimina il bucket. Se vuoi soltanto svuotare il bucket, accertati di avere rimosso l'istruzione che lo elimina.

Per ulteriori informazioni sull'utilizzo di altri AWS SDK, consulta Tools for Amazon Web Services.

Utilizzo di una configurazione del ciclo di vita

Per svuotare un bucket di grandi dimensioni, ti consigliamo di utilizzare una regola di configurazione del ciclo di vita S3. La scadenza del ciclo di vita è un processo asincrono, pertanto l'esecuzione della regola potrebbe richiedere alcuni giorni prima che il bucket sia vuoto. Dopo la prima volta che Amazon S3 esegue la regola, tutti gli oggetti idonei alla scadenza vengono contrassegnati per l'eliminazione. Non vengono più addebitati costi per gli oggetti contrassegnati per l'eliminazione. Per ulteriori informazioni, consulta Come posso svuotare un bucket Amazon S3 utilizzando una regola di configurazione del ciclo di vita?.

Se si utilizza una configurazione del ciclo di vita per svuotare il bucket, tale configurazione deve includere versioni correnti e non correnti, contrassegni di eliminazione e caricamenti in più parti incompleti.

È possibile aggiungere le regole di configurazione del ciclo di vita per predisporre la scadenza di tutti gli oggetti o di un sottogruppo degli stessi con uno specifico prefisso nel nome della chiave. Ad esempio, per eliminare tutti gli oggetti in un bucket, è possibile impostare una regola del ciclo di vita per predisporre la scadenza degli oggetti il giorno successivo alla creazione degli stessi.

Amazon S3 supporta una regola per il ciclo di vita del bucket che può essere utilizzata per interrompere i caricamenti multiparti che non sono stati completati entro un determinato numero di giorni dopo l'avvio. Si consiglia di configurare questa regola del ciclo di vita per ridurre al minimo i costi di storage. Per ulteriori informazioni, consulta Configurazione del ciclo di vita del bucket per l'eliminazione dei caricamenti in più parti incompleti.

Per ulteriori informazioni sull'utilizzo di una configurazione del ciclo di vita per svuotare un bucket, consulta Impostazione di una configurazione del ciclo di vita su un bucket e Oggetti in scadenza.

Svuotare un secchio con configurato AWS CloudTrail

AWS CloudTrail tiene traccia degli eventi relativi ai dati a livello di oggetto in un bucket Amazon S3, come l'eliminazione di oggetti. Se utilizzi un bucket come destinazione per registrare i tuoi CloudTrail eventi e stai eliminando oggetti dallo stesso bucket, potresti creare nuovi oggetti mentre svuoti il bucket. Per evitare che ciò accada, interrompi i tuoi percorsi. AWS CloudTrail Per ulteriori informazioni su come impedire ai CloudTrail percorsi di registrare gli eventi, consulta Disattivazione della registrazione di un percorso nella Guida per l'AWS CloudTrail utente.

Un'altra alternativa per impedire che i CloudTrail percorsi vengano aggiunti al bucket consiste nell'aggiungere una dichiarazione di `s3:PutObject` negazione alla policy relativa ai bucket. Se desideri memorizzare nuovi oggetti nel bucket in un secondo momento, dovrai rimuovere questa istruzione di negazione `s3:PutObject`. Per ulteriori informazioni, consulta Operazioni sugli oggetti ed Elementi delle policy JSON IAM: Effect nella Guida per l'utente IAM.

Eliminazione di un bucket

Un bucket Amazon S3 vuoto può essere eliminato. Prima di eliminare un bucket, considera quanto segue:

- I nomi dei bucket sono univoci. Se elimini un bucket, un altro AWS utente può utilizzare il nome.
- Se il bucket ospita un sito Web statico e hai creato e configurato una zona ospitata di Amazon Route 53 come descritto in Tutorial: Configurazione di un sito Web statico utilizzando un dominio personalizzato registrato con Route 53, devi ripulire le impostazioni della zona ospitata di Route 53 associate al bucket. Per ulteriori informazioni, consulta Passaggio 2: eliminare la zona ospitata Route 53.
- Se il bucket riceve i dati di log da Elastic Load Balancing (ELB): consigliamo di interrompere la consegna dei log ELB al bucket prima dell'eliminazione. Dopo l'eliminazione del bucket, se un altro utente crea un bucket utilizzando lo stesso nome, i dati di log potrebbero potenzialmente essere consegnati a quel bucket. Per informazioni sui registri di accesso ELB, consulta le sezioni Registri di accesso nella Guida per l'utente di Classic Load Balancer e Registri di accesso nella Guida per l'utente di Application Load Balancer.

Risoluzione dei problemi

Se non riesci a eliminare un bucket Amazon S3, considera quanto segue:

- Assicurati che il bucket sia vuoto – Puoi eliminare solo bucket che non hanno oggetti al loro interno. Assicurati che il bucket sia vuoto.
- Assicurati che non siano presenti punti di accesso collegati: puoi eliminare solo bucket a cui non sono collegati punti di accesso. Elimina tutti i punti di accesso collegati al bucket, prima di eliminare il bucket.
- AWS Organizations policy di controllo del servizio (SCP): una policy di controllo del servizio può negare l'autorizzazione di eliminazione su un bucket. Per informazioni sulle SCP, consulta Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .
- s3: DeleteBucket permessi — Se non riesci a eliminare un bucket, contatta il tuo amministratore IAM per confermare di disporre delle autorizzazioni. `s3:DeleteBucket` Per informazioni su come visualizzare o aggiornare le autorizzazioni IAM, consulta la sezione Modifica delle autorizzazioni per un utente IAM nella Guida per l'utente IAM.
- s3: DeleteBucket deny statement — Se disponi di `s3:DeleteBucket` autorizzazioni nella tua policy IAM e non puoi eliminare un bucket, la policy del bucket potrebbe includere un'istruzione di negazione per. `s3:DeleteBucket` Per impostazione predefinita, i bucket creati da ElasticBeanstalk hanno una politica contenente questa dichiarazione. Prima di poter eliminare il bucket, è necessario eliminare questa istruzione o la policy del bucket.

> **Important**
>
> I nomi dei bucket sono univoci. Se elimini un bucket, un altro AWS utente può utilizzare il nome. Se desideri continuare a utilizzare lo stesso nome di bucket, non eliminare il bucket. Ti consigliamo di eliminare solo il contenuto del bucket.

Utilizzo della console S3

Per eliminare un bucket S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) selezionare l'opzione accanto al nome del bucket che si desidera eliminare, quindi scegliere Delete (Elimina) nella parte superiore della pagina.
3. Nella pagina Delete bucket (Elimina bucket) confermare che si desidera eliminare il bucket immettendo il nome del bucket nel campo di testo e quindi scegliere Delete bucket (Elimina bucket).

> **Note**
>
> Se il bucket contiene oggetti, svuotarlo prima di eliminarlo selezionando il collegamento di configurazione bucket vuoto nell'avviso di errore This bucket not empty (Questo bucket non è vuoto) e seguendo le istruzioni nella pagina Empty bucket (Svuota bucket). Quindi tornare alla pagina Delete bucket (Elimina bucket) ed eliminare il bucket.

4. Per verificare di aver eliminato il bucket, apri l'elenco Bucket e inerisci il nome del bucket eliminato. Se il bucket non appare tra i risultati, la cancellazione si è conclusa correttamente.

Utilizzo dell' AWS SDK for Java

L'esempio seguente mostra come eliminare un bucket utilizzando l' AWS SDK for Java. In primo luogo, il codice elimina gli oggetti presenti nel bucket e quindi il bucket stesso. Per informazioni su altri SDK AWS , consulta Strumenti per Amazon Web Services.

Java

Nell'esempio Java seguente viene eliminato un bucket che contiene oggetti. Tale codice elimina tutti gli oggetti e quindi il bucket stesso. L'esempio di codice vale sia per i bucket che supportano la funzione Controllo delle versioni che per quelli che non la supportano.

> **Note**
>
> Per i bucket che non supportano la funzione Controllo delle versioni, è possibile eliminare direttamente tutti gli oggetti e poi il bucket stesso. Per i bucket che supportano la funzione Controllo delle versioni, è necessario eliminare tutte le versioni degli oggetti prima di eliminare il bucket.

Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella Developer Guide. 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();
        }
    }
}
```

Utilizzo del AWS CLI

Puoi eliminare un bucket che contiene oggetti AWS CLI se non ha il controllo delle versioni abilitato. Quando elimini un bucket che contiene oggetti, tutti gli oggetti nel bucket vengono eliminati definitivamente, inclusi gli oggetti che sono passati alla classe di storage `S3 Glacier`.

Se il tuo bucket non ha il controllo delle versioni abilitato, puoi utilizzare il AWS CLI comando `rb` (remove bucket) con il `--force` parametro per eliminare il bucket e tutti gli oggetti in esso contenuti. Questo comando elimina prima tutti gli oggetti e quindi il bucket stesso.

Se il controllo delle versioni è abilitato, gli oggetti con versione non verranno eliminati in questo processo: ciò comporterebbe il fallimento dell'eliminazione del bucket in quanto non vuoto. Per ulteriori informazioni sull'eliminazione di oggetti con versione, consulta la sezione Eliminazione delle versioni degli oggetti.

```
$ aws s3 rb s3://bucket-name --force
```

Per ulteriori informazioni, consulta Uso dei comandi S3 di alto livello con la nella Guida per l'utente. AWS Command Line Interface AWS Command Line Interface

Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva negli SDK and. AWS Command Line Interface AWS Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

La crittografia è configurata per tutti i bucket Amazon S3 per impostazione predefinita; gli oggetti vengono crittografati automaticamente utilizzando la crittografia lato server con chiavi gestite da

Amazon S3 (SSE-S3). Questa impostazione di crittografia si applica a tutti gli oggetti nei bucket Amazon S3.

Se hai bisogno di un maggiore controllo sulle tue chiavi, come la gestione della rotazione delle chiavi e le concessioni delle policy di accesso, puoi scegliere di utilizzare la crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS) o la crittografia lato server a due livelli con AWS KMS chiavi (DSSE-KMS). Per ulteriori informazioni sulla modifica delle chiavi KMS, consulta Modifica delle chiavi nella Guida per gli sviluppatori di AWS Key Management Service .

> **Note**
>
> Abbiamo modificato i bucket per crittografare automaticamente i caricamenti di nuovi oggetti. Se in precedenza hai creato un bucket senza crittografia predefinita, Amazon S3 abiliterà la crittografia per impostazione predefinita per il bucket utilizzando SSE-S3. Non verranno apportate modifiche alla configurazione della crittografia predefinita per un bucket con chiavi SSE-S3 o SSE-KMS già configurate. Per crittografare gli oggetti con SSE-KMS, è necessario modificare il tipo di crittografia nelle impostazioni del bucket. Per ulteriori informazioni, consulta Utilizzo della crittografia lato server con chiavi (SSE-KMS) AWS KMS.

Quando configuri il bucket per utilizzare la crittografia predefinita con SSE-KMS, puoi anche abilitare S3 Bucket Keys per ridurre il traffico delle richieste da Amazon S3 e ridurre il costo della crittografia. AWS KMS Per ulteriori informazioni, consulta Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3.

Per identificare i bucket in cui è abilitato SSE-KMS per la crittografia predefinita, puoi utilizzare i parametri di Amazon S3 Storage Lens. S3 Storage Lens è una funzionalità di analisi dell'archiviazione su cloud che puoi utilizzare per avere una panoramica completa a livello di organizzazione sull'utilizzo e sulle attività relative all'archiviazione di oggetti. Per ulteriori informazioni, consulta Utilizzo di S3 Storage Lens per proteggere i dati.

Quando utilizzi la crittografia lato server, Amazon S3 esegue la crittografia di un oggetto prima di salvarlo su disco e lo decritta al momento del download. Per ulteriori informazioni sulla protezione dei dati mediante la crittografia lato server e la gestione delle chiavi di crittografia, consulta Protezione dei dati con la crittografia lato server.

Per ulteriori informazioni sulle autorizzazioni richieste per la crittografia predefinita, consulta PutBucketEncryption nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Puoi configurare il comportamento di crittografia predefinito di Amazon S3 per un bucket S3 utilizzando la console Amazon S3, gli SDK AWS , l'API REST di Amazon S3 e l'interfaccia a riga di comando (). AWS AWS CLI

Crittografia di oggetti esistenti

Per crittografare gli oggetti Amazon S3 non crittografati esistenti, puoi utilizzare la funzionalità Operazioni in batch Amazon S3. Fornisci alle operazioni in batch S3 un elenco di oggetti da utilizzare e le operazioni in batch chiamano la rispettiva API per eseguire l'operazione specifica. È possibile utilizzare l'operazione di copia delle operazioni in batch per copiare gli oggetti non crittografati esistenti e scrivere i nuovi oggetti crittografati nello stesso bucket. Un solo processo di operazioni in batch può eseguire l'operazione specificata su miliardi di oggetti. Per ulteriori informazioni, consulta Esecuzione di operazioni in batch su larga scala su oggetti Amazon S3 e il post del Blog sull'archiviazione di AWS Crittografia di oggetti Amazon S3 esistenti con le operazioni in batch di Amazon S3.

Puoi anche crittografare gli oggetti esistenti utilizzando l'operazione API o il comando. `CopyObject` `copy-object` AWS CLI Per ulteriori informazioni, consulta il post del Blog sull'archiviazione di AWS Crittografia di oggetti Amazon S3 esistenti con AWS CLI l.

> **Note**
>
> I bucket Amazon S3 con la crittografia predefinita SSE-KMS non possono essere utilizzati come bucket di destinazione per the section called “Registrazione dell'accesso al server”. Solo la crittografia predefinita SSE-S3 è supportata per i bucket di destinazione del log di accesso server.

Utilizzo della crittografia SSE-KMS per operazioni multi-account

Quando si utilizza la crittografia per operazioni multi-account, tieni presente quanto segue:

- Se non viene fornito un AWS KMS key Amazon Resource Name (ARN) o un alias al momento della richiesta o tramite la configurazione di crittografia predefinita del bucket, viene utilizzata la Chiave gestita da AWS ()`aws/s3`.
- Se stai caricando o accedendo a oggetti S3 utilizzando principi AWS Identity and Access Management (IAM) che sono gli stessi Account AWS della tua chiave KMS, puoi usare il (). Chiave gestita da AWS `aws/s3`

- Se desideri concedere l'accesso multi-account agli oggetti S3, utilizza una chiave gestita dal cliente. Puoi configurare la policy di una chiave gestita dal cliente per consentire l'accesso da un altro account.
- Se stai specificando una chiave KMS gestita dal cliente, ti consigliamo di utilizzare una chiave KMS ARN completamente qualificata. Se invece utilizzi un alias di chiave KMS, AWS KMS risolve la chiave all'interno dell'account del richiedente. Ciò potrebbe comportare la crittografia dei dati con una chiave KMS di proprietà del richiedente e non del proprietario del bucket.
- È necessario specificare una chiave per cui il richiedente ha ottenuto l'autorizzazione `Encrypt`. Per ulteriori informazioni, consulta l'argomento relativo all'autorizzazione concessa agli utenti delle chiavi di utilizzare una chiave KMS per le operazioni di crittografia nella Guida per gli sviluppatori di AWS Key Management Service .

Per ulteriori informazioni su quando utilizzare le chiavi gestite dal cliente e le chiavi KMS AWS gestite, consulta Devo usare una chiave Chiave gestita da AWS o una chiave gestita dal cliente per crittografare i miei oggetti in Amazon S3?

Utilizzo della codifica predefinita con la replica

Una volta abilitata la crittografia predefinita per un bucket di destinazione della replica, si applica il seguente comportamento di crittografia:

- Se gli oggetti nel bucket di origine non sono crittografati, gli oggetti replicati nel bucket di destinazione vengono crittografati in base alle impostazioni di crittografia predefinita del bucket di destinazione. Di conseguenza, i tag di entità (ETag) degli oggetti di origine differiscono dagli ETag degli oggetti di replica. Se disponi di applicazioni che utilizzano ETag, devi aggiornarle per tenere conto di questa differenza.
- Se gli oggetti nel bucket di origine sono crittografati utilizzando la crittografia lato server con chiavi gestite Amazon S3 (SSE-S3), la crittografia lato server con chiavi () (SSE-KMS AWS KMS) o la crittografia lato server a doppio livello con AWS Key Management Service AWS KMS chiavi (DSSE-KMS), gli oggetti di replica nel bucket di destinazione utilizzano lo stesso tipo di crittografia degli oggetti di origine. Le impostazioni della crittografia predefinita del bucket di destinazione non vengono utilizzate.

Per ulteriori informazioni sull'utilizzo della crittografia di default con SSE-KMS, consulta Replica di oggetti crittografati.

Utilizzo di chiavi bucket Amazon S3 con crittografia predefinita

Quando configuri il bucket per utilizzare SSE-KMS come funzionalità di crittografia predefinita per i nuovi oggetti, puoi anche configurare le chiavi bucket S3. Le S3 Bucket Keys riducono il numero di transazioni da Amazon S3 AWS KMS per ridurre il costo di SSE-KMS.

Quando configuri il bucket per utilizzare S3 Bucket Keys per SSE-KMS su nuovi oggetti, AWS KMS genera una chiave a livello di bucket che viene utilizzata per creare una chiave dati univoca per gli oggetti nel bucket. Questa S3 Bucket Key viene utilizzata per un periodo di tempo limitato all'interno di Amazon S3, riducendo la necessità per Amazon S3 di effettuare richieste per completare le operazioni di crittografia. AWS KMS

Per ulteriori informazioni sull'utilizzo delle chiavi del bucket S3, consulta la sezione Utilizzo di chiavi bucket Amazon S3.

Configurazione della crittografia predefinita

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva negli SDK and. AWS Command Line Interface AWS Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

I bucket Amazon S3 hanno la crittografia dei bucket abilitata per impostazione predefinita; i nuovi oggetti vengono crittografati automaticamente utilizzando la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3). Questa crittografia si applica a tutti i nuovi oggetti nei bucket Amazon S3 e non comporta costi aggiuntivi.

Se hai bisogno di un maggiore controllo sulle chiavi di crittografia, come la gestione della rotazione delle chiavi e le concessioni delle policy di accesso, puoi scegliere di utilizzare la crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS) o la crittografia lato server a due livelli con AWS KMS chiavi (DSSE-KMS). Per ulteriori informazioni su SSE-KMS, consulta

Specifica della crittografia lato server con AWS KMS (SSE-KMS). Per ulteriori informazioni su DSSE-KMS, consulta the section called “Crittografia lato server a doppio livello (DSSE-KMS)”.

Se desideri utilizzare una chiave KMS di proprietà di un account diverso, devi avere l'autorizzazione necessaria per l'uso della chiave. Per ulteriori informazioni sulle autorizzazioni tra account per le chiavi KMS, vedi Creazione di chiavi KMS utilizzabili da altri account nella Guida per gli sviluppatori di AWS Key Management Service .

Quando imposti la crittografia dei bucket predefinita su SSE-KMS, puoi anche configurare una S3 Bucket Key per ridurre i costi delle richieste. AWS KMS Per ulteriori informazioni, consulta Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3.

> **Note**
>
> Se utilizzi PutBucketEncryptionla crittografia dei bucket predefinita su SSE-KMS, devi verificare che l'ID della tua chiave KMS sia corretto. Amazon S3 non convalida l'ID della chiave KMS fornito nelle richieste. PutBucketEncryption

L'uso della crittografia predefinita dei bucket S3 non comporta costi aggiuntivi. Per le richieste di configurare la funzione di crittografia predefinita vengono applicati i costi standard per le richieste Amazon S3. Per informazioni sui prezzi, consulta Prezzi di Amazon S3. Per SSE-KMS e DSSE-KMS, vengono applicati dei costi indicati nella tabella dei prezzi. AWS KMSAWS KMS

La crittografia lato server con chiavi di crittografia fornite dal cliente (SSE-C) è supportata per la crittografia predefinita.

Puoi configurare la crittografia predefinita di Amazon S3 per un bucket S3 utilizzando la console Amazon S3, gli SDK AWS , l'API REST di Amazon S3 e (). AWS Command Line Interface AWS CLI

Modifiche alla nota prima dell'abilitazione della crittografia predefinita

Una volta abilitata la crittografia predefinita di un bucket, si applica il seguente comportamento di crittografia:

- Non avvengono modifiche della crittografia degli oggetti che esisteva nel bucket prima che la crittografia predefinita venisse abilitata.
- Quando si effettua il caricamento di oggetti dopo l'abilitazione della crittografia predefinita:
  - Se le intestazioni della richiesta PUT non includono le informazioni di crittografia, Amazon S3 utilizza le impostazioni di crittografia di default del bucket per eseguire la crittografia degli oggetti.

- Se le intestazioni della richiesta PUT includono le informazioni di crittografia, Amazon S3 utilizza le informazioni di crittografia della richiesta PUT per eseguire la crittografia degli oggetti prima di archiviarli in Amazon S3.
- Se usi l'opzione SSE-KMS o DSSE-KMS per la configurazione della crittografia predefinita, vengono applicati i limiti di richieste al secondo (RPS) di AWS KMS. Per ulteriori informazioni sulle quote AWS KMS e su come richiedere un aumento delle quote, consulta Quote nella Guida per gli sviluppatori di AWS Key Management Service .

> **Note**
>
> Gli oggetti caricati prima dell'abilitazione della crittografia predefinita non verranno crittografati. Per ulteriori informazioni sulla crittografia di oggetti, consulta the section called "Impostazione della crittografia predefinita del bucket".

Utilizzo della console S3

Per configurare la crittografia predefinita per un bucket Amazon S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket scegli il nome del bucket desiderato.
4. Scegliere la scheda Properties (Proprietà).
5. In Default encryption (Crittografia di default), scegliere Edit (Modifica).
6. Per configurare la crittografia, in Tipo di crittografia scegli una delle seguenti opzioni:
   - Crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3)
   - Crittografia lato server con AWS Key Management Service chiavi (SSE-KMS)
   - Crittografia lato server a doppio livello con chiavi (DSSE-KMS) AWS Key Management Service

   > **Important**
   >
   > Se usi l'opzione SSE-KMS o DSSE-KMS per la configurazione della crittografia predefinita, vengono applicati i limiti di richieste al secondo (RPS) di AWS KMS. Per

ulteriori informazioni sulle AWS KMS quote e su come richiedere un aumento delle quote, consulta Quotas nella Developer Guide.AWS Key Management Service

I bucket e i nuovi oggetti sono crittografati per impostazione predefinita con SSE-S3, a meno che non specifichi un altro tipo di crittografia predefinita per i bucket. Per ulteriori informazioni sulla crittografia predefinita, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3.

Per ulteriori informazioni sull'utilizzo della crittografia lato server di Amazon S3 per crittografare i dati, consulta Uso della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).

7. Se hai scelto la crittografia lato server con AWS Key Management Service chiavi (SSE-KMS) o la crittografia lato server a doppio livello con chiavi (DSSE-KMS), procedi come segue: AWS Key Management Service

   a. In Chiave AWS KMS specifica la tua chiave KMS in uno dei seguenti modi:

      - Per scegliere da un elenco di chiavi KMS disponibili, scegli tra le tue e scegli la tua AWS KMS keys chiave KMS dall'elenco delle chiavi disponibili.

        In questo elenco vengono visualizzate sia la chiave Chiave gestita da AWS (`aws/s3`) che quella gestita dai clienti. Per ulteriori informazioni sulle chiavi gestite dai clienti, consulta Customer keys and AWS keys nella AWS Key Management Service Developer Guide.
      - Per specificare l'ARN della chiave KMS, scegli Inserisci l'ARN della AWS KMS key e quindi specifica l'ARN della chiave KMS nel campo visualizzato.
      - Per creare una nuova chiave gestita dal cliente nella AWS KMS console, scegli Crea una chiave KMS.

        Per ulteriori informazioni sulla creazione di una AWS KMS key, consulta Creating keys nella AWS Key Management Service Developer Guide.

      Important

      Puoi usare solo chiavi KMS abilitate nello Regione AWS stesso bucket. Quando scegli Choose from your KMS keys (Scegli tra le chiavi KMS), la console S3 elenca solo 100 chiavi KMS per regione. Se hai più di 100 chiavi KMS nella stessa regione, puoi vedere solo le prime 100 chiavi KMS nella console S3. Per utilizzare una

chiave KMS non elencata nella console, seleziona Inserisci l'ARN AWS KMS key e specifica l'ARN della chiave KMS.
Quando utilizzi una chiave KMS AWS KMS key per la crittografia lato server in Amazon S3, devi scegliere una chiave KMS di crittografia simmetrica. Amazon S3 supporta solo chiavi KMS di crittografia simmetrica. Per ulteriori informazioni sulle chiavi, consulta Chiavi KMS di crittografia simmetrica nella Guida per gli sviluppatori di AWS Key Management Service .

Per ulteriori informazioni sull'uso di SSE-KMS con Amazon S3, consulta Utilizzo della crittografia lato server con chiavi (SSE-KMS) AWS KMS. Per ulteriori informazioni sull'uso di DSSE-KMS, consulta the section called “Crittografia lato server a doppio livello (DSSE-KMS)”.

b. Quando si configura il bucket per utilizzare la crittografia predefinita con SSE-KMS, è anche possibile abilitare le chiavi bucket S3. S3 Bucket Keys riduce il costo della crittografia diminuendo il traffico di richieste da Amazon S3 a. AWS KMS Per ulteriori informazioni, consulta Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3.

   Per utilizzare le chiavi bucket S3, in Chiave bucket seleziona Abilita.

   > **Note**
   >
   > Le chiavi bucket S3 non sono supportate per DSSE-KMS.

8. Seleziona Salvataggio delle modifiche.

Usando il AWS CLI

Questi esempi mostrano come configurare la crittografia predefinita utilizzando la crittografia gestita da Amazon S3 (SSE-S3) o la crittografia SSE-KMS con una chiave bucket S3.

Per ulteriori informazioni sulla crittografia predefinita, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3. Per ulteriori informazioni sull'utilizzo della AWS CLI configurazione della crittografia predefinita, vedere put-bucket-encryption.

Example - Crittografia predefinita con SSE-S3

In questo esempio viene configurata la crittografia predefinita dei bucket con le chiavi gestite da Amazon S3.

```
aws s3api put-bucket-encryption --bucket example-s3-bucket --server-side-encryption-
configuration '{
    "Rules": [
        {
            "ApplyServerSideEncryptionByDefault": {
                "SSEAlgorithm": "AES256"
            }
        }
    ]
}'
```

Example - Crittografia predefinita con SSE-KMS utilizzando una chiave bucket S3

In questo esempio viene configurata la crittografia predefinita del bucket con SSE-KMS utilizzando una chiave bucket S3.

```
aws s3api put-bucket-encryption --bucket example-s3-bucket --server-side-encryption-
configuration '{
    "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "aws:kms",
                    "KMSMasterKeyID": "KMS-Key-ARN"
                },
                "BucketKeyEnabled": true
            }
        ]
    }'
```

Utilizzo di REST API

Usa l'operazione REST API `PutBucketEncryption` per abilitare la crittografia predefinita e impostare il tipo di crittografia lato server da utilizzare: SSE-S3, SSE-KMS o DSSE-KMS.

Per ulteriori informazioni, consulta PutBucketEncryption in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Monitoraggio della crittografia predefinita con AWS CloudTrail e Amazon EventBridge

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva negli SDK and. AWS Command Line Interface AWS Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

È possibile tenere traccia le richieste di configurazione della crittografia predefinita per i bucket Amazon S3 mediante gli eventi AWS CloudTrail . I seguenti nomi di eventi API vengono utilizzati nei log: CloudTrail

- `PutBucketEncryption`
- `GetBucketEncryption`
- `DeleteBucketEncryption`

Puoi anche creare EventBridge regole che corrispondano agli CloudTrail eventi per queste chiamate API. Per ulteriori informazioni sugli CloudTrail eventi, consultaAbilitazione della registrazione per gli oggetti in un bucket utilizzando la console. Per ulteriori informazioni sugli EventBridge eventi, vedere Events from Servizi AWS.

Puoi utilizzare CloudTrail i log per le azioni Amazon S3 a livello di oggetto per tracciare `PUT` e inviare richieste ad Amazon S3. `POST` È possibile utilizzare queste azioni per verificare se la crittografia predefinita viene utilizzata per crittografare gli oggetti quando le richieste `PUT` in arrivo non dispongono di intestazioni di crittografia.

Quando Amazon S3 esegue la crittografia di un oggetto in base alle impostazioni della crittografia predefinita, il log include uno dei seguenti campi come coppia nome/valore:

`"SSEApplied":"Default_SSE_S3"`, `"SSEApplied":"Default_SSE_KMS"` o `"SSEApplied":"Default_DSSE_KMS"`.

Quando Amazon S3 esegue la crittografia di un oggetto in base alle intestazioni di crittografia PUT, il log include uno dei campi seguenti come coppia nome/valore: `"SSEApplied":"SSE_S3"`, `"SSEApplied":"SSE_KMS"`, `"SSEApplied":"DSSE_KMS"` o `"SSEApplied":"SSE_C"`.

Per i caricamenti in più parti, queste informazioni sono incluse nelle richieste dell'operazione API `InitiateMultipartUpload`. Per ulteriori informazioni sull'utilizzo di and, consulta. CloudTrail CloudWatch Monitoraggio di Amazon S3

Lavorare con Mountpoint per Amazon S3

Mountpoint per Amazon S3 è un client di file open source ad alta velocità di trasmissione effettiva per il montaggio di un bucket Amazon S3 come file system locale. Con Mountpoint, le applicazioni possono accedere agli oggetti archiviati in Amazon S3 tramite operazioni sui file system, come apertura e lettura. Mountpoint converte automaticamente queste operazioni in chiamate API a oggetti S3, offrendo alle applicazioni l'accesso all'archiviazione elastica e alla velocità di trasmissione effettiva di Amazon S3 tramite un'interfaccia di file.

Mountpoint per Amazon S3 è generalmente disponibile per l'uso in produzione su applicazioni su larga scala con carico di lavoro in lettura elevato: data lake, formazione di machine learning, rendering di immagini, simulazione di veicoli autonomi, estrazione, trasformazione e caricamento (ETL) e altro ancora.

Mountpoint supporta le operazioni di base del file system e può leggere file di dimensioni fino a 5 TB. Può elencare e leggere file esistenti e crearne di nuovi. Non può modificare file esistenti o eliminare directory e non supporta collegamenti simbolici o il blocco dei file. Mountpoint è ideale per le applicazioni che non necessitano di tutte le funzionalità di un file system condiviso e di autorizzazioni in stile POSIX, ma richiedono il velocità di trasmissione effettiva elastica di Amazon S3 per leggere e scrivere set di dati S3 di grandi dimensioni. Per i dettagli, consulta Mountpoint file system behavior su GitHub. Per carichi di lavoro che richiedono il supporto POSIX completo, consigliamo Amazon FSx per Lustre e il relativo supporto per il collegamento di bucket S3.

Mountpoint per Amazon S3 è disponibile solo per sistemi operativi Linux. Puoi utilizzare Mountpoint per accedere agli oggetti S3 in tutte le classi di archiviazione ad eccezione di Recupero flessibile Amazon S3 Glacier, Deep Archive Amazon S3 Glacier, livello S3 Intelligent-Tiering Archive Access e livello S3 Intelligent-Tiering Deep Archive Access.

Argomenti

Installazione di Mountpoint

Puoi scaricare e installare pacchetti predefiniti di Mountpoint per Amazon S3 utilizzando la riga di comando. Le istruzioni per scaricare e installare Mountpoint variano a seconda del sistema operativo Linux che stai utilizzando.

Argomenti

Distribuzioni basate su RPM (Amazon Linux, Fedora, CentOS, RHEL)

1. Copia il seguente URL di download per la tua architettura.

   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. Scarica il pacchetto Mountpoint per Amazon S3. Sostituisci *download-link* con l'URL di download appropriato della fase precedente.

   ```
   wget download-link
   ```

3. (Facoltativo) Verifica dell'integrità e dell'autenticità dei file scaricati. Per prima cosa, copia l'URL della firma appropriato per la tua architettura.

   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
```

Quindi, consulta Verifica della firma del pacchetto Mountpoint per Amazon S3.

4. Installa il pacchetto utilizzando il seguente comando:

```
sudo yum install ./mount-s3.rpm
```

5. Verifica che Mountpoint sia installato correttamente inserendo il seguente comando:

```
mount-s3 --version
```

Verrà visualizzato un output simile al seguente:

```
mount-s3 1.3.1
```

Distribuzioni basate su DEB (Debian, Ubuntu)

1. Copia l'URL di download per la tua architettura.

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. Scarica il pacchetto Mountpoint per Amazon S3. Sostituisci *download-link* con l'URL di download appropriato della fase precedente.

```
wget download-link
```

3. (Facoltativo) Verifica dell'integrità e dell'autenticità dei file scaricati. Innanzitutto, copia l'URL della firma per la tua architettura.

   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
   ```

   Quindi, consulta Verifica della firma del pacchetto Mountpoint per Amazon S3.

4. Installa il pacchetto utilizzando il seguente comando:

   ```
   sudo apt-get install ./mount-s3.deb
   ```

5. Verifica che Mountpoint per Amazon S3 sia installato correttamente eseguendo il seguente comando:

   ```
   mount-s3 --version
   ```

   Verrà visualizzato un output simile al seguente:

   ```
   mount-s3 1.3.1
   ```

Altre distribuzioni Linux

1. Consulta la documentazione del sistema operativo per installare i pacchetti `FUSE` e `libfuse2`, che sono obbligatori.
2. Copia l'URL di download per la tua architettura.

   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. Scarica il pacchetto Mountpoint per Amazon S3. Sostituisci *download-link* con l'URL di download appropriato della fase precedente.

```
wget download-link
```

4. (Facoltativo) Verifica dell'integrità e dell'autenticità dei file scaricati. Innanzitutto, copia l'URL della firma per la tua architettura.

   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
```

   Quindi, consulta Verifica della firma del pacchetto Mountpoint per Amazon S3.

5. Installa il pacchetto utilizzando il seguente comando:

```
sudo mkdir -p /opt/aws/mountpoint-s3 && sudo tar -C /opt/aws/mountpoint-s3 -xzf ./
mount-s3.tar.gz
```

6. Aggiungi il file binario `mount-s3` alla variabile di ambiente `PATH`. Nel file `$HOME/.profile`, aggiungi la seguente riga:

```
export PATH=$PATH:/opt/aws/mountpoint-s3/bin
```

   Salva il file `.profile` ed esegui il seguente comando:

```
source $HOME/.profile
```

7. Verifica che Mountpoint per Amazon S3 sia installato correttamente eseguendo il seguente comando:

```
mount-s3 --version
```

Verrà visualizzato un output simile al seguente:

```
mount-s3 1.3.1
```

Verifica della firma del pacchetto Mountpoint per Amazon S3

1. Installa GnuPG (il comando gpg). È necessario verificare l'autenticità e l'integrità di un pacchetto Mountpoint per Amazon S3 scaricato. GnuPG è installato per impostazione predefinita sulle Amazon Machine Images (AMI) di Amazon Linux. Dopo l'installazione di GnuPG, passa alla fase 2.
2. Scarica la chiave pubblica Mountpoint eseguendo il seguente comando:

   ```
   wget https://s3.amazonaws.com/mountpoint-s3-release/public_keys/KEYS
   ```
3. Importa la chiave pubblica Mountpoint nel keyring eseguendo il seguente comando:

   ```
   gpg --import KEYS
   ```
4. Verifica l'impronta digitale della chiave pubblica Mountpoint eseguendo il seguente comando:

   ```
   gpg --fingerprint mountpoint-s3@amazon.com
   ```

   Verifica che la stringa di impronte digitali visualizzata corrisponda a quanto segue:

   ```
   673F E406 1506 BB46 9A0E  F857 BE39 7A52 B086 DA5A
   ```

   Se la stringa di impronte digitali non corrisponde, non terminare l'installazione di Mountpoint e contatta AWS Support.
5. Scarica il file SIGNATURE del pacchetto. Sostituisci *signature-link* con l'apposito link per la firma riportato nelle sezioni precedenti.

   ```
   wget signature-link
   ```
6. Verifica la firma del pacchetto scaricato eseguendo il seguente comando. Sostituisci *signature-filename* con il nome del file della fase precedente.

```
gpg --verify signature-filename
```

Ad esempio, per distribuzioni basate su RPM, incluso Amazon Linux, esegui il comando seguente:

```
gpg --verify mount-s3.rpm.asc
```

7. L'output deve includere la frase `Good signature`. Se l'output include la frase `BAD signature`, scarica nuovamente il file del pacchetto Mountpoint e ripeti queste fasi. Se il problema persiste, non terminare l'installazione di Mountpoint e contatta AWS Support.

   L'output può includere un avviso relativo a una firma attendibile. Ciò non indica un problema. Significa solo che non hai verificato in modo indipendente la chiave pubblica Mountpoint.

Configurazione e utilizzo di Mountpoint

Per utilizzare Mountpoint per Amazon S3, il tuo host necessita di credenziali AWS valide con accesso al bucket o ai bucket che desideri montare. Per le diverse modalità di autenticazione, consulta Mountpoint AWS Credentials su GitHub.

Ad esempio, puoi creare un nuovo utente e ruolo AWS Identity and Access Management (IAM) per questo scopo. Assicurati che questo ruolo abbia accesso al bucket o ai bucket che desideri montare. Puoi passare il ruolo IAM all'istanza Amazon EC2 con un profilo dell'istanza.

Utilizzo di Mountpoint per Amazon S3

Usa Mountpoint per Amazon S3 per effettuare le seguenti operazioni:

1. Montare i bucket con il comando `mount-s3`.

   Nell'esempio seguente, sostituisci *DOC-EXAMPLE-BUCKET* con il nome del bucket S3 e sostituisci *~/mnt* con la directory sull'host in cui desideri che venga montato il bucket S3.

   ```
   mkdir ~/mnt
   mount-s3 DOC-EXAMPLE-BUCKET ~/mnt
   ```

   Poiché il client Mountpoint viene eseguito in background per impostazione predefinita, la directory *~/mnt* ora fornisce l'accesso agli oggetti nel bucket S3.

2. Accesso agli oggetti nel bucket tramite Mountpoint.

   Dopo aver montato il bucket localmente, puoi utilizzare comandi Linux comuni, come `cat` o `ls`, per lavorare con gli oggetti S3. Mountpoint per Amazon S3 interpreta le chiavi nel bucket S3 come percorsi di file system suddividendole sul carattere barra (`/`). Ad esempio, se disponi della chiave oggetto `Data/2023-01-01.csv` nel bucket, nel file system Mountpoint avrai una directory denominata `Data`, con un file denominato `2023-01-01.csv` al suo interno.

   Mountpoint per Amazon S3 non implementa intenzionalmente la specifica standard completa POSIX per i file system. Mountpoint è ottimizzato per carichi di lavoro che richiedono un accesso in lettura e scrittura con elevata velocità di trasmissione effettiva ai dati archiviati in Amazon S3 tramite un'interfaccia di file system, ma che per il resto non si basano sulle funzionalità del file system. Per ulteriori informazioni, consulta Mountpoint for Amazon S3 file system behavior su GitHub. I clienti che necessitano di una semantica del file system più ricca dovrebbero prendere in considerazione altri servizi di AWS file, come Amazon Elastic File System (Amazon EFS) o Amazon FSx.

3. Smontaggio del bucket usando il comando `umount`. Questo comando smonta il bucket S3 ed esce da Mountpoint.

   Per utilizzare il comando di esempio seguente, sostituisci *`~/mnt`* con la directory sull'host in cui è montato il bucket S3.

   ```
   umount ~/mnt
   ```

   > **Note**
   >
   > Per ottenere un elenco di opzioni per questo comando, eseguiumount `--help`.

Per ulteriori dettagli sulla configurazione di Mountpoint, consulta S3 bucket configuration e file system configuration su GitHub.

Configurazione della memorizzazione nella cache in Mountpoint

Quando usi Mountpoint per Amazon S3, puoi configurarlo per memorizzare nella cache i dati a cui hai effettuato l'accesso più di recente dai bucket S3 nell'archiviazione di istanze Amazon EC2 o in un volume Amazon EBS collegato. La memorizzazione nella cache di questi dati può aiutarti ad

accelerare le prestazioni e ridurre i costi dell'accesso ripetuto ai dati. La memorizzazione nella cache in Mountpoint è ideale per i casi d'uso in cui si leggono ripetutamente gli stessi dati che non cambiano durante le letture multiple. Ad esempio, puoi utilizzare la memorizzazione nella cache per le attività di training di machine learning che richiedono una lettura ripetuta di un set di dati di training per migliorare l'accuratezza del modello.

Quando monti un bucket S3, puoi facoltativamente abilitare la memorizzazione nella cache tramite flag. Puoi configurare la posizione e le dimensioni della cache dei dati e la quantità di tempo in cui i metadati vengono mantenuti nella cache. Quando monti un bucket e la memorizzazione nella cache è abilitata, Mountpoint crea una sottodirectory vuota nella posizione della cache configurata, se quella sottodirectory non esiste già. Quando monti un bucket per la prima volta e quando lo smonti, Mountpoint elimina il contenuto della posizione della cache. Per ulteriori informazioni sulla configurazione e l'uso della memorizzazione nella cache in Mountpoint, consulta Mountpoint for Amazon S3 Caching configuration on. GitHub

Quando monti un bucket S3, puoi facoltativamente abilitare la memorizzazione nella cache tramite flag `--cache` *`CACHE_PATH`*. Nell'esempio seguente, sostituisci *`CACHE_PATH`* con il percorso file della directory in cui desideri memorizzare i dati nella cache. Nell'esempio seguente, sostituisci *`DOC-EXAMPLE-BUCKET`* con il nome del bucket S3 e sostituisci *`~/mnt`* con la directory sull'host in cui desideri che venga montato il bucket S3.

```
mkdir ~/mnt
mount-s3 --cache CACHE_PATH DOC-EXAMPLE-BUCKET ~/mnt
```

> **Important**
>
> Se abiliti la memorizzazione nella cache, Mountpoint mantiene il contenuto degli oggetti non crittografati dal bucket S3 nella posizione della cache configurata al momento del montaggio. Per proteggere i dati, è opportuno limitare l'accesso alla posizione della cache dei dati.

Risoluzione dei problemi di Mountpoint

Mountpoint per Amazon S3 è supportato da. AWS Support Se hai bisogno di assistenza, contatta il AWS Support Center.

Puoi anche esaminare e inviare Problemi relativi a Mountpoint su GitHub.

Se scopri un potenziale problema di sicurezza in questo progetto, ti chiediamo di segnalarlo ad AWS Security tramite la pagina di segnalazione delle vulnerabilità. Non creare un problema GitHub pubblico.

Se l'applicazione si comporta in modo imprevisto con Mountpoint, è possibile controllare le informazioni dei log per diagnosticare il problema.

Registrazione di log

Per impostazione predefinita, Mountpoint emette informazioni di log ad elevata gravità per syslog.

Per visualizzare i log sulla maggior parte delle distribuzioni Linux moderne, incluso Amazon Linux, esegui il seguente comando `journald`:

```
journalctl -e SYSLOG_IDENTIFIER=mount-s3
```

Su altri sistemi Linux, le voci `syslog` sono probabilmente scritte in un file come `/var/log/syslog`.

Puoi utilizzare questi log per risolvere i problemi dell'applicazione. Ad esempio, se l'applicazione tenta di sovrascrivere un file esistente, l'operazione non riesce e nel log verrà visualizzata una riga simile alla seguente:

```
[WARN] open{req=12 ino=2}: mountpoint_s3::fuse: open failed: inode error: inode 2 (full
 key "README.md") is not writable
```

Per ulteriori informazioni, consulta Mountpoint for Amazon S3 Logging su GitHub.

Configurazione di trasferimenti veloci e sicuri di file con Amazon S3 Transfer Acceleration

Amazon S3 Transfer Acceleration è una funzionalità a livello di bucket che permette il trasferimento rapido, semplice e sicuro di file su lunga distanza tra un client e un bucket S3. Transfer Acceleration è progettato per ottimizzare le velocità di trasferimento da tutto il mondo verso i bucket S3. Transfer Acceleration sfrutta le edge location distribuite a livello globale di Amazon CloudFront. Quando arrivano in una edge location, i dati vengono instradati ad Amazon S3 su un percorso di rete ottimizzato.

Quando si utilizza Transfer Acceleration, potrebbero essere applicati costi aggiuntivi per il trasferimento dei dati. Per ulteriori informazioni sui prezzi, consulta la sezione Prezzi di Amazon S3.

Perché utilizzare Transfer Acceleration?

L'utilizzo di Transfer Acceleration in un bucket è consigliabile in diversi casi:

- Clienti che effettuano il caricamento in un bucket centralizzato da ogni parte del mondo.
- Trasferimento regolare da qualche gigabyte a diversi terabyte di dati tra vari continenti.
- Durante il caricamento su Amazon S3 non è possibile utilizzare tutta la larghezza di banda disponibile su Internet.

Per maggiori informazioni sui casi in cui utilizzare Transfer Acceleration, consulta le Domande frequenti su Amazon S3.

Requisiti per l'utilizzo di Transfer Acceleration

Di seguito sono riportati i requisiti per l'utilizzo di Transfer Acceleration in un bucket S3:

- Transfer Acceleration è supportato solo in caso di richieste in stile hosting virtuale. Per ulteriori informazioni sulle richieste in stile hosting virtuale, consulta Esecuzione di richieste con l'utilizzo di API REST.
- Il nome del bucket utilizzato per Transfer Acceleration deve essere conforme a DNS e non deve contenere punti (".").
- Transfer Acceleration deve essere abilitato nel bucket. Per ulteriori informazioni, consulta Abilitazione e utilizzo di S3 Transfer Acceleration.

  Dopo avere abilitato Transfer Acceleration in un bucket, la velocità di trasferimento dei dati verso il bucket aumenta nel giro di 20 minuti.

  > **Note**
  >
  > La funzionalità Accelerazione del trasferimento attualmente non è supportata per i bucket situati nelle seguenti regioni:
  >
  > - Asia Pacifico (Tokyo) (ap-northeast-1)
  > - Asia Pacifico (Seoul) (ap-northeast-2)
  > - Asia Pacifico (Mumbai) (ap-south-1)
  > - Asia Pacifico (Singapore) (ap-southeast-1)
  > - Asia Pacifico (Sydney) (ap-southeast-2)

- Canada (Centrale) (ca-central-1)
- Europa (Francoforte) (eu-central-1)
- Europa (Irlanda) (eu-west-1)
- Europa (Londra) (eu-west-2)
- Europe (Parigi) (eu-west-3)
- Sud America (San Paolo) (sa-east-1)
- Stati Uniti orientali (Virginia settentrionale) (us-east-1)
- Stati Uniti orientali (Ohio) (us-east-2)
- Stati Uniti occidentali (California settentrionale) (us-west-1)
- Stati Uniti occidentali (Oregon) (us-west-2)

- Per accedere al bucket abilitato per Transfer Acceleration, è necessario utilizzare l'endpoint *bucketname*`.s3-accelerate.amazonaws.com`. In alternativa, puoi utilizzare l'endpoint dual-stack *bucketname*`.s3-accelerate.dualstack.amazonaws.com` per connetterti al bucket abilitato su IPv6. Puoi continuare a utilizzare gli endpoint normali per il trasferimento di dati standard.
- Per impostare lo stato di Transfer Acceleration, è necessario essere il proprietario del bucket. Il proprietario del bucket può assegnare autorizzazioni ad altri utenti in modo che possano impostare lo stato di accelerazione nel bucket. L'autorizzazione `s3:PutAccelerateConfiguration` consente agli utenti di attivare o disattivare Transfer Acceleration in un bucket. L'`s3:GetAccelerateConfiguration`autorizzazione consente agli utenti di restituire lo stato Transfer Acceleration di un bucket, che è o `Enabled` `Suspended`.

Le sezioni seguenti descrivono come iniziare e utilizzare Amazon S3 Transfer Acceleration per il trasferimento dei dati.

Argomenti

- Nozioni di base su Amazon S3 Transfer Acceleration
- Abilitazione e utilizzo di S3 Transfer Acceleration
- Utilizzo dello strumento Speed Comparison di Amazon S3 Transfer Acceleration

Nozioni di base su Amazon S3 Transfer Acceleration

È possibile utilizzare Amazon S3 Transfer Acceleration per il trasferimento rapido, semplice e sicuro di file su lunga distanza tra un client e un bucket S3. Transfer Acceleration utilizza le edge location distribuite a livello globale di Amazon CloudFront. Quando arrivano in una edge location, i dati vengono instradati ad Amazon S3 su un percorso di rete ottimizzato.

Per iniziare a utilizzare Amazon S3 Transfer Acceleration, eseguire le fasi descritte di seguito:

1. Attivazione di Transfer Acceleration su un bucket

   È possibile abilitare Transfer Acceleration in un bucket in uno dei seguenti modi:

   - Utilizzare la console di Amazon S3
   - Utilizza l'operazione REST API PUT Bucket accelerate.
   - Usa gli AWS SDK AWS CLI e. Per ulteriori informazioni, consulta Sviluppo con Amazon S3 utilizzando gli SDK AWS.

   Per ulteriori informazioni, consultare Abilitazione e utilizzo di S3 Transfer Acceleration.

   > **Note**
   >
   > Per consentire il funzionamento del bucket con Transfer Acceleration, il nome del bucket deve essere conforme ai requisiti di denominazione DNS e non deve contenere punti (".").

2. Trasferimento dei dati da e verso il bucket abilitato per l'accelerazione

   Utilizza uno dei seguenti nomi di dominio endpoint s3-accelerate:

   - Per accedere a un bucket abilitato per l'accelerazione, utilizza *`bucketname`*`.s3-accelerate.amazonaws.com`.
   - Per accedere a un bucket abilitato per l'accelerazione su IPv6, utilizza *`bucketname`*`.s3-accelerate.dualstack.amazonaws.com`.

     Gli endpoint dual-stack Amazon S3; supportano le richieste ai bucket S3 su IPv6 e IPv4. L'endpoint dual-stack Transfer Acceleration utilizza solo il tipo di nome di endpoint in stile hosting virtuale. Per ulteriori informazioni, consulta Nozioni di base sull'esecuzione di richieste su IPv6 e Utilizzo degli endpoint dual-stack Amazon S3.

> **Note**
>
> L'applicazione di trasferimento dei dati deve utilizzare uno dei due tipi di endpoint seguenti per accedere al bucket per il trasferimento dati rapido: `.s3-accelerate.amazonaws.com` o `.s3-accelerate.dualstack.amazonaws.com` per l'endpoint dual-stack. Se desideri utilizzare il trasferimento di dati standard, puoi continuare a utilizzare gli endpoint normali.

È possibile indirizzare le richieste PUT object e GET object di Amazon S3 sul nome di dominio endpoint `s3-accelerate` dopo avere abilitato Transfer Acceleration. Ad esempio, si supponga di disporre attualmente di un'applicazione REST API che utilizza PUT Object che utilizza il nome host `mybucket.s3.us-east-1.amazonaws.com` nella richiesta PUT. Per accelerare PUT, si modifica il nome host nella richiesta in `mybucket.s3-accelerate.amazonaws.com`. Per tornare a utilizzare la velocità di caricamento standard, modifica nuovamente il nome in `mybucket.s3.us-east-1.amazonaws.com`.

Una volta abilitato Transfer Acceleration, sarà possibile riscontrare miglioramenti delle prestazioni nel giro di 20 minuti. Tuttavia, l'endpoint di accelerazione sarà disponibile non appena viene abilitato Transfer Acceleration.

Puoi utilizzare l'acceleratore endpoint negli AWS CLI AWS SDK e in altri strumenti che trasferiscono dati da e verso Amazon S3. Se utilizzi gli AWS SDK, alcune delle lingue supportate utilizzano un flag di configurazione del client di accelerazione degli endpoint, quindi non è necessario impostare esplicitamente l'endpoint su Transfer Acceleration. *`bucketname`*`.s3-accelerate.amazonaws.com` Per gli esempi su come utilizzare un flag di configurazione del client per l'endpoint di accelerazione, consulta Abilitazione e utilizzo di S3 Transfer Acceleration.

Puoi utilizzare tutte le operazioni di Amazon S3 negli endpoint di accelerazione del trasferimento, ad eccezione di quanto segue:

- GET Service (elenco bucket)
- PUT Bucket (crea bucket)
- DELETE Bucket

Inoltre, Amazon S3 Transfer Acceleration non supporta le copie tra regioni mediante l'utilizzo di PUT Object - Copy.

Abilitazione e utilizzo di S3 Transfer Acceleration

Puoi utilizzare Amazon S3 Transfer Acceleration per trasferire i file in modo rapido e sicuro su lunghe distanze tra il tuo client e un bucket S3. Puoi abilitare Transfer Acceleration utilizzando la console S3, il AWS Command Line Interface (AWS CLI), l'API o gli SDK. AWS

In questa sezione vengono forniti alcuni esempi di come abilitare Amazon S3 Transfer Acceleration in un bucket e utilizzare l'endpoint di accelerazione per il bucket abilitato.

Per ulteriori informazioni sui requisiti di Transfer Acceleration, consulta Configurazione di trasferimenti veloci e sicuri di file con Amazon S3 Transfer Acceleration.

Utilizzo della console S3

> **Note**
>
> Se desideri confrontare le velocità di caricamento accelerate e non accelerate, apri lo strumento Speed Comparison di Amazon S3 Transfer Acceleration.
> Lo strumento Speed Comparison utilizza il caricamento in più parti per trasferire un file dal browser a vari file Regioni AWS con e senza l'accelerazione di trasferimento di Amazon S3. Puoi confrontare la velocità di caricamento per i caricamenti diretti e trasferire i caricamenti accelerati per Regione.

Per abilitare Transfer Acceleration per un bucket S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket, scegliere il nome del bucket per il quale si vuole abilitare Transfer Acceleration.
3. Scegli Properties (Proprietà).
4. In Transfer acceleration (Accelerazione trasferimento), scegliere Edit (Modifica).
5. Scegliere Enable (Abilita) e quindi Save changes (Salva modifiche).

Per accedere a trasferimenti di dati accelerati

1. Dopo che Amazon S3 ha attivato Transfer Acceleration per il bucket, consulta la scheda Proprietà del bucket.
2. In Transfer acceleration, Endpoint accelerated (Accelerated endpoint) visualizza l'endpoint Transfer acceleration per il bucket. Utilizza questo endpoint per accedere ai trasferimenti accelerati di dati da e verso il bucket.

   Sospendendo Transfer Acceleration, l'endpoint dell'accelerazione non funziona più.

Usando il AWS CLI

Di seguito sono riportati alcuni esempi di AWS CLI comandi utilizzati per Transfer Acceleration. Per istruzioni sulla configurazione di AWS CLI, vedereSviluppo con Amazon S3 tramite la AWS CLI.

Attivazione di Transfer Acceleration su un bucket

Utilizzate il AWS CLI put-bucket-accelerate-configurationcomando per abilitare o sospendere Transfer Acceleration su un bucket.

Nell'esempio che segue `Status=Enabled` viene impostato per l'abilitazione di Transfer Acceleration in un bucket. Viene utilizzato `Status=Suspended` per sospendere Transfer Acceleration.

Example

```
$ aws s3api put-bucket-accelerate-configuration --bucket bucketname --accelerate-
configuration Status=Enabled
```

Utilizzo di Transfer Acceleration

Puoi indirizzare tutte le richieste Amazon S3 effettuate dai AWS CLI comandi s3 e s3api all'endpoint di accelerazione:. `s3-accelerate.amazonaws.com` Per fare ciò, imposta il valore `use_accelerate_endpoint` di configurazione su `true` in un profilo nel tuo file AWS Config. Per utilizzare l'endpoint di accelerazione, è necessario che Transfer Acceleration sia abilitato nel bucket.

Tutte le richieste vengono inviate tramite il modello di indirizzamento virtuale del bucket: `my-bucket.s3-accelerate.amazonaws.com`. Qualsiasi richiesta `ListBuckets`, `CreateBucket` e `DeleteBucket` non verrà inviata all'endpoint di accelerazione in quanto tale endpoint non supporta queste operazioni.

Per ulteriori informazioni su `use_accelerate_endpoint`, consulta Configurazione di AWS CLI S3 in Guida di riferimento dei comandi AWS CLI .

Nell'esempio che segue, `use_accelerate_endpoint` viene impostato su `true` nel profilo di default.

Example

```
$ aws configure set default.s3.use_accelerate_endpoint true
```

Se desideri utilizzare l'endpoint di accelerazione per alcuni AWS CLI comandi ma non per altri, puoi utilizzare uno dei due metodi seguenti:

- Puoi utilizzare l'endpoint di accelerazione per qualsiasi comando s3 o s3api impostando il parametro `--endpoint-url` su `https://s3-accelerate.amazonaws.com`.
- Imposta profili separati nel tuo file AWS Config. Ad esempio, si può creare un profilo che imposta `use_accelerate_endpoint` su `true` e un profilo che non imposta `use_accelerate_endpoint`. Quando si esegue un comando, specifica il profilo da usare, a seconda dell'intenzione di utilizzare o meno l'endpoint di accelerazione.

Caricamento di un oggetto in un bucket abilitato per Transfer Acceleration

Nell'esempio che segue viene caricato un file in un bucket abilitato per Transfer Acceleration mediante il profilo predefinito configurato per l'utilizzo dell'endpoint di accelerazione.

Example

```
$ aws s3 cp file.txt s3://bucketname/keyname --region region
```

Nell'esempio che segue viene caricato un file in un bucket abilitato per Transfer Acceleration mediante il parametro `--endpoint-url` per specificare l'endpoint di accelerazione.

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
```

Utilizzo degli SDK AWS

Di seguito sono riportati alcuni esempi di utilizzo di Transfer Acceleration per caricare oggetti su Amazon S3 utilizzando l' AWS SDK. *Alcuni dei linguaggi supportati dall' AWS SDK (ad esempio, Java e.NET) utilizzano un flag di configurazione del client di accelerazione degli endpoint, quindi non è necessario impostare esplicitamente l'endpoint per Transfer Acceleration su bucketname .s3-accelerate.amazonaws.com.*

Java

Example

Nell'esempio seguente viene mostrato come utilizzare un endpoint di accelerazione per il caricamento di un oggetto in Amazon S3. Inoltre, vengono effettuate le seguenti operazioni:

- Viene creato un `AmazonS3Client` configurato per utilizzare un endpoint di accelerazione. Tutti i bucket cui accede il client devono avere Transfer Acceleration abilitato.
- Abilita Transfer Acceleration in un bucket specificato. Questa fase è necessaria solo se sul bucket specificato non è ancora abilitato Transfer Acceleration.
- Viene verificato se Transfer Acceleration è abilitato per il bucket specificato.
- Viene caricato un nuovo oggetto nel bucket specificato utilizzando l'endpoint di accelerazione del bucket.

Per ulteriori informazioni sull'uso di Transfer Acceleration, consulta Nozioni di base su Amazon S3 Transfer Acceleration. Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella Developer Guide. 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

L'esempio seguente mostra come utilizzare per AWS SDK for .NET abilitare l'accelerazione del trasferimento su un bucket. Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

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);
                }
        }
    }
}
```

Durante il caricamento di un oggetto in un bucket con Transfer Acceleration abilitato, specifica l'utilizzo dell'endpoint di accelerazione durante la creazione di un client.

```
var client = new AmazonS3Client(new AmazonS3Config
            {
                RegionEndpoint = TestRegionEndpoint,
                UseAccelerateEndpoint = true
            }
```

Javascript

Per un esempio di abilitazione dell'accelerazione del trasferimento utilizzando l' AWS SDK per JavaScript, consulta Calling the putBucketAccelerate Configuration operation in the AWS SDK for API Reference. JavaScript

Python (Boto)

Per un esempio di come abilitare Transfer Acceleration mediante l'utilizzo di SDK per Python, consulta put_bucket_accelerate_configuration nella Documentazione di riferimento dell'API SDK AWS per Python (Boto3).

Other

Per informazioni sull'utilizzo di altri AWS SDK, consulta Codice di esempio e librerie.

Utilizzo di REST API

Utilizza l'operazione REST API `PutBucketAccelerateConfiguration` per abilitare la configurazione accelerata su un bucket esistente.

Per ulteriori informazioni, consulta il riferimento PutBucketAccelerateConfigurationall'API di Amazon Simple Storage Service.

Utilizzo dello strumento Speed Comparison di Amazon S3 Transfer Acceleration

Puoi utilizzare lo strumento Speed Comparison di Amazon S3 Transfer Acceleration per confrontare le velocità di caricamento accelerate e non accelerate tra regioni Amazon S3. Lo strumento Speed Comparison usa caricamenti in più parti per trasferire un file dal browser a diverse regioni Amazon S3 con o senza l'utilizzo di Transfer Acceleration.

È possibile accedere allo strumento Speed Comparison utilizzando uno dei metodi riportati di seguito:

- Copia il seguente URL nella finestra del browser, sostituendo la *`regione`* con Regione AWS quella che stai utilizzando (ad esempio`us-west-2`) e *`yourBucketName`*con il nome del bucket che desideri valutare:

  ```
  https://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/
  accelerate-speed-comparsion.html?
  region=region&origBucketName=yourBucketName
  ```

  Per un elenco delle regioni supportate da Amazon S3, consultare la sezione relativa a endpoint e quote di Amazon S3 nella Riferimenti generali di AWS.
- Utilizzare la console di Amazon S3

Utilizzo dei bucket con pagamento a carico del richiedente per utilizzo e trasferimenti di storage

In generale, i proprietari dei bucket pagano tutti i costi di storage e trasferimento dei dati Amazon S3 associati al loro bucket. Tuttavia, è possibile configurare un bucket in modo che sia un bucket con pagamento a carico del richiedente. Nel caso di bucket con Pagamento a carico del richiedente, il costo della richiesta e del download dei dati dal bucket viene pagato dal richiedente anziché dal proprietario del bucket. Il proprietario del bucket paga sempre il costo di archiviazione dei dati.

Generalmente, si configurano bucket Requester Pays quando si desidera condividere dati senza incorrere nei costi associati all'accesso ai dati da parte di altre persone. Ad esempio, puoi utilizzare bucket con pagamento a carico del richiedente se desideri rendere disponibili grandi set di dati, come directory di codici postali, dati di riferimento, informazioni geospaziali o dati di Web crawling.

> **Important**
>
> Se si abilita il Pagamento a carico del richiedente su un bucket, l'accesso anonimo a quel bucket non è consentito.

È necessario autenticare tutte le richieste che riguardano i bucket con Pagamento a carico del richiedente. L'autenticazione delle richieste consente ad Amazon S3 di identificare il richiedente e addebitargli l'utilizzo del bucket con Pagamento a carico del richiedente.

Quando il richiedente assume un ruolo AWS Identity and Access Management (IAM) prima di effettuare la richiesta, l'account a cui appartiene il ruolo viene addebitato per la richiesta. Per ulteriori informazioni sui ruoli IAM, consultare Ruoli IAM nella Guida per l'utente di IAM.

Dopo aver configurato un bucket come bucket Requester Pays, i richiedenti devono dimostrare di aver compreso che verranno addebitati i costi per la richiesta e il download dei dati. Per dimostrare di accettare gli addebiti, i richiedenti devono includere `x-amz-request-payer` come intestazione nella richiesta API per le richieste DELETE, GET, HEAD, POST e PUT oppure aggiungere il `RequestPayer` parametro nella richiesta REST. Per le richieste CLI, i richiedenti possono utilizzare il parametro. `--request-payer`

Example — Utilizzo di Requester Pays durante l'eliminazione di un oggetto

Per utilizzare il seguente esempio di DeleteObjectVersionAPI, sostituiscilo *`user input placeholders`* con le tue informazioni.

```
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
```

Se il richiedente ripristina gli oggetti utilizzando l'RestoreObjectAPI, Requester Pays è supportato purché l'`x-amz-request-payer`intestazione o il `RequestPayer` parametro siano presenti nella richiesta; tuttavia, il richiedente paga solo il costo della richiesta. Il proprietario del bucket paga i costi di recupero.

I bucket con pagamento a carico del richiedente non supportano quanto riportato di seguito.

- Richieste anonime
- Richieste SOAP
- L'uso di un bucket con pagamento a carico del richiedente come bucket di destinazione per la registrazione degli utenti finali o viceversa. Tuttavia, è possibile attivare la registrazione degli utenti finale su un bucket con pagamento a carico del richiedente in cui il bucket di destinazione non è un bucket di questo genere.

Come funzionano i pagamenti a carico del richiedente

L'addebito delle richieste di pagamento a carico del richiedente che hanno esisto positivo è diretto: il richiedente paga il trasferimento dei dati e la richiesta; il proprietario del bucket paga lo storage dei dati. Tuttavia, il proprietario del bucket riceve l'addebito della richiesta nei casi seguenti:

- La richiesta restituisce un errore `AccessDenied` (HTTP`403 Forbidden`) e viene avviata all'interno dell'account o dell'organizzazione individuale del proprietario del bucket. AWS AWS
- La richiesta è una richiesta SOAP.

Per ulteriori informazioni sui pagamenti a carico del richiedente, consulta gli argomenti riportati di seguito.

Argomenti

- Configurazione di pagamenti a carico del richiedente su un bucket

- Recupero della configurazione requestPayment tramite REST API
- Scaricamento di oggetti dai bucket Requester Pays

Configurazione di pagamenti a carico del richiedente su un bucket

Puoi configurare un bucket Amazon S3 in modo che sia un bucket con pagamento a carico del richiedente in modo che il richiedente paghi il costo della richiesta e il download dei dati al posto del proprietario del bucket.

In questa sezione sono riportati esempi di come configurare i pagamenti a carico del richiedente per un un bucket Amazon S3 utilizzando la console e REST API.

Utilizzo della console S3

Per abilitare il Pagamento a carico del richiedente per un bucket S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket scegliere il nome del periodo fisso per il quale si vuole abilitare il Pagamento a carico del richiedente.
3. Scegliere Properties (Proprietà).
4. In Requester pays (Pagamento a carico del richiedente), scegliere Edit (Modifica).
5. Scegliere Enable (Abilita) e quindi Save changes (Salva modifiche).

   Amazon S3 abilita il Pagamento a carico del richiedente per il bucket e visualizza la panoramica del bucket. In Pagamento a carico del richiedente si può notare che è Abilitato.

Utilizzo dell'API REST

Solo il proprietario del bucket può impostare il valore di configurazione `RequestPaymentConfiguration.payer` di un bucket su `BucketOwner`, impostazione predefinita, o su `Requester`. La configurazione della risorsa `requestPayment` è facoltativa. Per impostazione predefinita, il bucket non è un bucket con Pagamento a carico del richiedente.

Per riportare il bucket con Pagamento a carico del richiedente a un bucket normale, si utilizza il valore `BucketOwner`. Generalmente, si utilizza `BucketOwner` quando si caricano dati nel bucket Amazon S3 e successivamente si imposta il valore su `Requester` prima di pubblicare gli oggetti nel bucket.

Per configurare requestPayment

- Utilizzare una richiesta PUT per impostare il valore `Payer` su `Requester` in un bucket specificato.

```
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>
```

Se la richiesta ha esito positivo, Amazon S3 restituisce una risposta simile a quella riportata di seguito.

```
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
```

Puoi impostare il pagamento a carico del richiedente solo a livello di bucket. Non è possibile impostare il pagamento a carico del richiedente per oggetti specifici all'interno del bucket.

È possibile configurare un bucket come `BucketOwner` o `Requester` in qualsiasi momento. Tuttavia, potrebbero essere necessari alcuni minuti prima che il nuovo valore di configurazione abbia effetto.

> **Note**
>
> I proprietari dei bucket che utilizzano URL prefirmati dovrebbero riflettere bene prima di configurare un bucket come bucket con pagamento a carico del richiedente, soprattutto se l'URL ha una durata di vita molto lunga. Il proprietario del bucket riceve l'addebito ogni volta che il richiedente utilizza un URL prefirmato associato alle credenziali del proprietario del bucket.

Recupero della configurazione requestPayment tramite REST API

È possibile determinare il valore `Payer` impostato in un bucket richiedendo la risorsa `requestPayment`.

Per ottenere la risorsa requestPayment

- Utilizzare una richiesta GET per ottenere la risorsa `requestPayment`, come mostrato nella richiesta seguente.

```
GET ?requestPayment HTTP/1.1
Host: [BucketName].s3.amazonaws.com
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]
```

Se la richiesta ha esito positivo, Amazon S3 restituisce una risposta simile a quella riportata di seguito.

```
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>
```

Questa risposta mostra che il valore `payer` è impostato su `Requester`.

Scaricamento di oggetti dai bucket Requester Pays

Poiché i richiedenti ricevono l'addebito del download dei dati dai bucket con pagamento a carico del richiedente, le richieste devono contenere un parametro speciale, `x-amz-request-payer`, che conferma che il richiedente sa che riceverà l'addebito del download. Per accedere agli oggetti nei

bucket con Pagamento a carico del richiedente, le richieste devono includere uno degli elementi seguenti.

- Per le richieste DELETE, GET, HEAD, POST e PUT, includere `x-amz-request-payer : requester` nell'intestazione
- Per gli URL firmati, includere `x-amz-request-payer=requester` nella richiesta

Se la richiesta ha esito positivo e il richiedente riceve l'addebito, la risposta include l'intestazione `x-amz-request-charged:requester`. Se la richiesta non contiene `x-amz-request-payer`, Amazon S3 restituisce un errore 403 e addebita la richiesta al proprietario del bucket.

> **Note**
>
> I proprietari dei bucket non devono aggiungere `x-amz-request-payer` alle loro richieste. Assicurarsi di aver incluso `x-amz-request-payer` e il suo valore nel calcolo della firma. Per ulteriori informazioni, vedete Costruzione dell'elemento. CanonicalizedAmzHeaders

Utilizzo di REST API

Per scaricare oggetti da un bucket con Pagamento a carico del richiedente

- Utilizzare una richiesta GET per scaricare un oggetto da un bucket con Pagamento a carico del richiedente, come mostrato nella richiesta seguente.

```
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]
```

Se la richiesta GET ha esito positivo e il richiedente riceve l'addebito, la risposta include `x-amz-request-charged:requester`.

Amazon S3 può restituire un errore `Access Denied` per le richieste di recupero di oggetti da un bucket con Pagamento a carico del richiedente. Per ulteriori informazioni, consulta Risposte agli errori nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Utilizzo del AWS CLI

Per scaricare oggetti da un bucket Requester Pays utilizzando il AWS CLI, specificate `--request-payer requester` come parte della richiesta`get-object`. Per ulteriori informazioni, consulta get-object nella Documentazione di riferimento della AWS CLI .

Restrizioni e limitazioni dei bucket

Un bucket Amazon S3 è di proprietà di chi lo Account AWS ha creato. La proprietà del bucket non è trasferibile ad un altro account.

Quando crei un bucket, ne scegli il nome e il nome in cui Regione AWS crearlo. Una volta creato, non potrai più modificarne il nome o la regione.

Quando assegni un nome a un bucket, dovrai scegliere un nome che sia rilevante per te o la tua attività. Evita di utilizzare nomi associati ad altri. Ad esempio, si dovrebbe evitare di utilizzare `AWS` o `Amazon` nel nome del bucket.

Per impostazione predefinita, puoi creare fino a 100 bucket in ciascuno dei tuoi. Account AWS Se necessiti di bucket aggiuntivi, puoi aumentare la quota di bucket dell'account fino a un massimo di 1.000 bucket inviando una richiesta di aumento della quota. Non ci sono differenze di prestazioni se si utilizzano molti o solo alcuni bucket.

> **Note**
>
> Non è necessario inviare più richieste di aumento della quota per ciascuna Regione AWS di esse. La quota del bucket viene applicata all' Account AWS.

Per informazioni su come aumentare la quota di bucket, consulta Quote dei servizi per AWS nei Riferimenti generali AWS .

Riutilizzo dei nomi dei bucket

Se un bucket è vuoto, puoi eliminarlo. Dopo l'eliminazione di un bucket, il nome diventa disponibile per un nuovo utilizzo. Tuttavia, dopo aver eliminato il bucket, potrebbe non essere possibile riutilizzare il nome per vari motivi.

Ad esempio, quando elimini il bucket e il nome diventa disponibile per il riutilizzo, un altro account Account AWS potrebbe creare un bucket utilizzando lo stesso nome. Inoltre, potrebbe essere

necessario qualche minuto prima di poter riutilizzare il nome di un bucket eliminato. Se desideri utilizzare lo stesso nome per il bucket, ti consigliamo di non eliminarlo affatto.

Per ulteriori informazioni sui nomi dei bucket, consulta Regole di denominazione dei bucket.

Limiti su oggetti e bucket

Non esiste alcun limite alle dimensioni massime del bucket o al numero di oggetti che è possibile archiviare in un bucket. È possibile archiviare tutti gli oggetti in un unico bucket oppure organizzarli in diversi bucket. Tuttavia, non puoi creare un bucket da un altro bucket.

Operazioni relative ai bucket

La progettazione ad alta disponibilità di Amazon S3 si basa sulle operazioni get, put, list e delete. Poiché le operazioni del bucket funzionano in uno spazio di risorse globale centralizzato, non è indicato creare, eliminare o configurare bucket nel percorso di codice ad alta disponibilità dell'applicazione. È preferibile creare, eliminare o configurare i bucket durante le distinte attività di routine di inizializzazione o di configurazione, che vengono eseguite con frequenza minore.

Denominazione e creazione automatica dei bucket

Se l'applicazione crea bucket automaticamente, scegli uno schema di denominazione dei bucket con meno probabilità di creare conflitti di denominazione. Assicurati che la logica dell'applicazione scelga un nome del bucket diverso nel caso in cui il nome del bucket sia già in uso.

Per ulteriori informazioni sulla denominazione dei bucket, consulta Regole di denominazione dei bucket.

Caricamento, download e utilizzo di oggetti in Amazon S3

Per memorizzare i tuoi dati in Amazon S3, lavori con risorse denominate bucket e oggetti. Un bucket è un container per oggetti o file. Un oggetto è un file e tutti i metadati che descrivono tale file.

Per memorizzare un oggetto in Amazon S3, crei un bucket e quindi carichi l'oggetto in un bucket. Quando l'oggetto si trova nel bucket, è possibile aprirlo, scaricarlo e copiarlo. Quando non hai più bisogno di un oggetto o di un bucket, puoi ripulire queste risorse.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

> **Important**
>
> Nella console di Amazon S3, quando scegli Open (Apri) o Download As (Scarica come) per un oggetto, queste operazioni creano URL prefirmati. L'oggetto sarà accessibile a chiunque abbia accesso a questi URL prefirmati per cinque minuti. Per ulteriori informazioni sugli URL prefirmati, consulta la sezione relativa all'utilizzo di URL prefirmati.

Con Amazon S3 paghi solo per le risorse utilizzate. Per ulteriori informazioni sulle funzionalità e sui prezzi di Amazon S3, consulta Amazon S3. Se sei un nuovo cliente Amazon S3, puoi iniziare a utilizzare Amazon S3 gratuitamente. Per ulteriori informazioni, consulta Piano gratuito AWS.

Argomenti

- Eliminazione di oggetti Amazon S3
- Organizzare, elencare e utilizzare gli oggetti
- Utilizzo di URL prefirmati
- Trasformazione di oggetti con S3 Object Lambda

Panoramica degli oggetti di Amazon S3

Amazon S3 è un object store che utilizza valori chiave univoci per archiviare tutti gli oggetti desiderati. Questi oggetti vengono archiviati in uno o più bucket e ogni oggetto può avere dimensioni fino a 5 TB. Un oggetto è costituito dai seguenti elementi:

Chiave

Il nome assegnato a un oggetto. La chiave dell'oggetto viene utilizzata per recuperare l'oggetto. Per ulteriori informazioni, consulta Utilizzo dei metadati degli oggetti.

ID versione

All'interno di un bucket, una chiave e l'ID versione identificano in modo univoco un oggetto. L'ID versione è una stringa generata da Amazon S3 quando aggiungi un oggetto a un bucket. Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Valore

Il contenuto che stai archiviando.

Il valore di un oggetto può essere costituito da qualsiasi sequenza di byte. La dimensione degli oggetti può essere compresa tra 0 e 5 TB. Per ulteriori informazioni, consulta Caricamento degli oggetti.

Metadati

Un set di coppie nome-valore con le quali è possibile archiviare le informazioni relative all'oggetto. È possibile assegnare metadati, denominati metadati definiti dall'utente, agli oggetti disponibili in Amazon S3. Inoltre, Amazon S3 assegna a questi oggetti metadati di sistema che utilizza per gestire gli oggetti. Per ulteriori informazioni, consulta Utilizzo dei metadati degli oggetti.

Risorse secondarie

Amazon S3 utilizza il meccanismo delle risorse secondarie per archiviare ulteriori informazioni specifiche dell'oggetto. Poiché le risorse secondarie sono subordinate agli oggetti, sono sempre

associate a un'altra entità, ad esempio un oggetto o un bucket. Per ulteriori informazioni, consulta Risorse secondarie degli oggetti.

Informazioni sul controllo degli accessi

È possibile controllare l'accesso agli oggetti archiviati in Amazon S3. Amazon S3 supporta sia il controllo degli accessi basato sulle risorse, ad esempio una lista di controllo degli accessi (ACL) e le policy dei bucket, oltre al controllo degli accessi basato sugli utenti. Per ulteriori informazioni sul controllo degli accessi, consulta:

- Gestione degli accessi
- Identity and Access Management per Amazon S3
- Configurazione delle ACL

Le risorse di Amazon S3 (ad esempio, bucket e oggetti) sono private per default. È necessario concedere l'autorizzazione in modo esplicito affinché altri utenti possano accedere alle risorse. Per ulteriori informazioni sulla condivisione degli oggetti, consulta Condivisione di oggetti mediante URL prefirmati.

Tag

È possibile utilizzare i tag per classificare gli oggetti archiviati, per il controllo degli accessi o l'allocazione dei costi. Per ulteriori informazioni, consulta Suddivisione in categorie dello storage utilizzando i tag.

Risorse secondarie degli oggetti

Amazon S3 definisce un set di risorse secondarie associate ai bucket e agli oggetti. Le risorse secondarie sono subordinate agli oggetti. Ciò significa che le risorse secondarie non esistono da sole. Sono sempre associate a qualche altra entità, ad esempio un oggetto o un bucket.

Nella tabella seguente sono elencate le risorse secondarie associate agli oggetti di Amazon S3.

Risorsa secondaria	Descrizione
acl	Contiene un elenco delle assegnazioni, che identificano gli assegnatari e le autorizzazioni concesse. Quando si crea un oggetto, la risorsa secondaria `acl` identifica il proprietario dell'oggetto assegnandogli il controllo completo sull'ogge tto. È possibile recuperare un'ACL dell'oggetto o sostituirla con un elenco aggiornat

Risorsa secondaria	Descrizione
	o delle assegnazioni. Qualsiasi aggiornamento apportato a un'ACL richiede la sostituzione dell'ACL esistente. Per ulteriori informazioni sulle ACL, consulta Panoramica delle liste di controllo accessi (ACL).

Creazione di nomi di chiavi oggetto

La chiave oggetto (o nome di chiave) identifica l'oggetto in modo univoco in un bucket Amazon S3. I metadati dell'oggetto sono invece un set di coppie nome-valore Per ulteriori informazioni sui metadati degli oggetti, consulta Utilizzo dei metadati degli oggetti.

Quando si crea un oggetto, specificare il nome della chiave che lo identifica in modo univoco nel bucket. Ad esempio, quando evidenzi un bucket nella console di Amazon S3, viene visualizzato un elenco degli oggetti nel bucket. Questi nomi sono le chiavi degli oggetti. Il nome della chiave dell'oggetto è una sequenza di caratteri Unicode con codifica UTF-8 di una lunghezza massima di 1.024 byte. I nomi delle chiavi degli oggetti fanno distinzione tra maiuscole e minuscole.

> **Note**
>
> I nomi delle chiavi degli oggetti con il valore «soap» non sono supportati per virtual-hosted-style le richieste. Per i valori dei nomi delle chiavi dell'oggetto in cui viene utilizzato "soap", deve invece essere usato un URL in stile percorso.

Il modello di dati di Amazon S3 è una struttura flat: crei un bucket e il bucket archivia gli oggetti. Non c'è nessuna gerarchia di bucket secondari o sottocartelle. Tuttavia, è possibile applicare una gerarchia logica utilizzando delimitatori e prefissi di nomi di chiavi come avviene nella console di Amazon S3. La console di Amazon S3 supporta il concetto di cartella. Per ulteriori informazioni su come modificare i metadati dalla console di Amazon S3, consulta Modifica dei metadati degli oggetti nella console di Amazon S3.

Supponiamo che il bucket (`admin-created`) contenga quattro oggetti con le seguenti chiavi:

```
Development/Projects.xls
```

```
Finance/statement1.pdf
```

`Private/taxdocument.pdf`

`s3-dg.pdf`

La console utilizza i prefissi dei nomi di chiavi (`Development/`, `Finance/` e `Private/`) e il delimitatore ('/') per visualizzare una struttura di cartelle. Poiché la chiave `s3-dg.pdf` non ha un prefisso, i relativi oggetti vengono visualizzati direttamente a livello root del bucket. Se si apre la cartella `Development/`, viene visualizzato l'oggetto `Projects.xlsx` in essa contenuto.

- Amazon S3 supporta i bucket e gli oggetti e non sono presenti gerarchie. Tuttavia, utilizzando prefissi e delimitatori in un nome chiave di oggetto, la console Amazon S3 e gli AWS SDK possono dedurre la gerarchia e introdurre il concetto di cartelle.
- La console di Amazon S3 implementa la creazione di oggetti cartella creando un oggetto a byte zero con il prefisso di cartella e il valore di delimitatore come chiave. Questi oggetti cartella non vengono visualizzati nella console. Altrimenti si comportano come qualsiasi altro oggetto e possono essere visualizzati e manipolati tramite l'API REST, la AWS CLI e gli SDK. AWS

Linee guida per la denominazione delle chiavi degli oggetti

Puoi utilizzare qualsiasi carattere UTF-8 all'interno del nome di un oggetto. Tuttavia, utilizzare alcuni caratteri nei nomi delle chiavi può causare problematiche con alcuni protocolli e applicazioni. Le seguenti linee guida garantiscono la massima conformità con DNS, i caratteri sicuri per il Web, i parser XML e altre interfacce API.

Caratteri sicuri

I seguenti set di caratteri possono essere utilizzati con la massima sicurezza nei nomi delle chiavi:

Caratteri alfanumerici	• 0-9 • a-z • A-Z
Caratteri speciali	• Punto esclamativo (`!`) • Trattino (`-`) • Carattere di sottolineatura (`_`) • Punto (`.`) • Asterisco (`*`)

- Virgoletta singola (')
- Parentesi aperta (()
- Parentesi chiusa ())

Di seguito sono riportati esempi di nomi di chiavi validi per gli oggetti:

- `4my-organization`
- `my.great_photos-2014/jan/myvacation.jpg`
- `videos/2014/birthday/video1.wmv`

> **Note**
>
> Gli oggetti con nomi chiave che terminano con punti "." scaricati utilizzando la console di Amazon S3 avranno i punti "." rimossi dal nome della chiave dell'oggetto scaricato. Per scaricare un oggetto il cui nome chiave termina con il punto (i) «.» conservato nell'oggetto scaricato, devi utilizzare AWS Command Line Interface (AWS CLI), gli AWS SDK o l'API REST.
>
> Inoltre, tieni a mente le seguenti limitazioni sui prefissi:
>
> - Gli oggetti con il prefisso «./» devono essere caricati o scaricati con AWS Command Line Interface (AWS CLI), AWS SDK o API REST. Non puoi utilizzare la console di Amazon S3.
> - Gli oggetti con il prefisso "../" non possono essere caricati utilizzando AWS Command Line Interface (AWS CLI) o la console di Amazon S3.

Caratteri che potrebbero richiedere una gestione speciale

I seguenti caratteri in un nome di chiave potrebbero richiedere ulteriori operazioni di gestione del codice e probabilmente dovranno essere codificati tramite URL o vi si dovrà fare riferimento come HEX. Alcuni di essi sono caratteri non stampabili ed è possibile che non vengano gestiti dal browser in uso; per tale motivo, richiedono una gestione speciale.

- E commerciale ("&")
- Dollaro ("$")
- I caratteri ASCII sono compresi tra 00-1F hex (0-31 decimale) e 7F (127 decimale)

- Simbolo "at" ("@")
- Uguale ("=")
- Punto e virgola (";")
- Barra obliqua ("/")
- Due punti (":")
- Più ("+")
- Spazio - È possibile che sequenze significative di spazi vadano perse in alcuni utilizzi (in particolare, gli spazi multipli)
- Virgola (",")
- Punto interrogativo ("?")

Caratteri da evitare

Ti consigliamo di non utilizzare i seguenti caratteri in un nome chiave a causa della notevole gestione dei caratteri speciali, che non è coerente in tutte le applicazioni.

- Barra rovesciata ("\")
- Parentesi graffa di apertura ("{")
- Caratteri ASCII non stampabili (caratteri decimali da 128 a 255)
- Accento circonflesso ("^")
- Parentesi graffa di chiusura ("}")
- Carattere di percentuale ("%")
- Accento grave/apice inverso ("`")
- Parentesi quadra di chiusura ("]")
- Virgolette
- Simbolo 'maggiore di' (">")
- Parentesi quadra di apertura ("[")
- Tilde ("~")
- Simbolo "minore di" ("<")
- Carattere "cancelletto" ("#")

- Barra verticale ("|")

Vincoli chiave degli oggetti correlati a XML

Come specificato dallo standard XML sulla end-of-line gestione, tutto il testo XML viene normalizzato in modo tale che le restituzioni a riga singola (codice ASCII 13) e le restituzioni a riga immediatamente seguite da un'alimentazione di riga (codice ASCII 10) vengano sostituite da un carattere di alimentazione a riga singola. Per garantire l'analisi corretta delle chiavi oggetto nelle richieste XML, i ritorni a capo e altri caratteri speciali devono essere sostituiti con il codice di entità XML equivalente quando vengono inseriti all'interno dei tag XML. Di seguito è riportato un elenco di tali caratteri speciali e dei loro codici di entità equivalenti:

- ' come `'`
- ” come `"`
- & come `&`
- < come `<`
- > come `>`
- \r come `
` o `
`
- \n come `
` o `
`

Example

Nell'esempio seguente viene illustrato l'utilizzo di un codice di entità XML come sostituzione di un ritorno a capo. Questa richiesta `DeleteObjects` elimina un oggetto con il parametro `key`: `/some/prefix/objectwith\rcarriagereturn` (dove \r è il ritorno a capo).

```
<Delete xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Object>
    <Key>/some/prefix/objectwith&#13;carriagereturn</Key>
  </Object>
</Delete>
```

Utilizzo dei metadati degli oggetti

Puoi impostare i metadati degli oggetti in Amazon S3 al momento del caricamento dell'oggetto. I metadati dell'oggetto sono invece un set di coppie nome-valore Una volta caricato l'oggetto, non è

possibile modificare i metadata corrispondenti. L'unico modo per modificarli è eseguire una copia dell'oggetto e impostare i metadata.

Quando si crea un oggetto, viene specificato anche il nome della chiave che lo identifica in modo univoco nel bucket. La chiave oggetto (o nome di chiave) identifica l'oggetto in modo univoco in un bucket Amazon S3. Per ulteriori informazioni, consulta Creazione di nomi di chiavi oggetto.

Esistono due tipi di metadati in Amazon S3: metadati definiti dal sistema e metadati definiti dall'utente. Le sezioni seguenti forniscono ulteriori informazioni sui metadati definiti dal sistema e definiti dall'utente. Per ulteriori informazioni sulla modifica dei metadati tramite la console di Amazon S3, consulta Modifica dei metadati degli oggetti nella console di Amazon S3.

Metadata di oggetti definiti dal sistema

Per ogni oggetto archiviato in un bucket, Amazon S3 mantiene un set di metadati di sistema. Questi metadati vengono elaborati da Amazon S3 in base alle necessità. Ad esempio, Amazon S3 mantiene i metadati sulla dimensione e sulla data di creazione degli oggetti e utilizza queste informazioni come parte della gestione degli oggetti.

Esistono due categorie di metadata di sistema:

- Controllati dal sistema: i metadati, come la data di creazione dell'oggetto, sono controllati dal sistema e il valore di questi metadati può essere modificato solo da Amazon S3.
- Controllati dall'utente: altri metadata di sistema, come la classe di storage configurata per l'oggetto e se la crittografia lato server è abilitata per l'oggetto, sono esempi di metadata di sistema, il cui valore viene controllato dall'utente. Se il bucket è configurato come sito Web, è possibile che si desideri reindirizzare la richiesta di una pagina a un'altra pagina o a un URL esterno. In questo caso, la pagina Web è un oggetto nel bucket. Amazon S3 archivia il valore di reindirizzamento della pagina come metadati di sistema, il cui valore può essere controllato dall'utente.

  Quando si creano oggetti, è possibile configurare i valori di questi metadata di sistema o aggiornarli in base alle esigenze. Per ulteriori informazioni sulle classi di storage, consulta Utilizzo delle classi di storage di Amazon S3.

  Amazon S3 utilizza AWS KMS le chiavi per crittografare gli oggetti Amazon S3. AWS KMS crittografa solo i dati dell'oggetto. Il checksum, insieme all'algoritmo specificato, vengono archiviati come parte dei metadati dell'oggetto. Se la crittografia lato server viene richiesta per l'oggetto, il checksum viene archiviato in formato crittografato. Per ulteriori informazioni sulla crittografia lato server, consulta Protezione dei dati con la crittografia.

> **Note**
>
> L'intestazione della richiesta PUT è limitata a una dimensione di 8 KB. Nell'intestazione della richiesta PUT, la dimensione dei metadati definiti dal sistema è limitata a 2 KB. La dimensione dei metadati definiti dal sistema viene calcolata sommando il numero di byte della codifica US-ASCII di ogni chiave e valore.

Nella tabella riportata di seguito viene fornito un elenco dei metadata definiti dal sistema e viene indicato se è possibile modificarli.

Nome	Descrizione	L'utente può modificare il valore?
`Date`	Data e ora correnti.	No
`Cache-Control`	Un campo di intestazione generico utilizzato per specifica re i criteri di memorizzazione nella cache.	Sì
`Content-D isposition`	Informazioni relative alla modalità di presentazione dell'oggetto.	Sì
`Content-Length`	Dimensioni dell'oggetto in byte.	No
`Content-Type`	Il tipo di oggetto.	Sì
`Last-Modified`	Data di creazione dell'oggetto o data dell'ultima modifica, scegliendo la più recente delle due. Per i caricamenti in più parti, la data di creazione dell'oggetto è la data di inizio del caricamento in più parti.	No
`ETag`	Tag di entità (ETag) che rappresenta la versione specifica di un oggetto. Per gli oggetti che non vengono caricati come caricamento in più parti e sono non crittografati oppure crittografati mediante crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3), l'ETag è un digest MD5 dei dati.	No

Nome	Descrizione	L'utente può modificare il valore?
`x-amz-server-side-encryption`	Un'intestazione che indica se la crittografia lato server è abilitata per l'oggetto e se tale crittografia utilizza le chiavi AWS Key Management Service (AWS KMS) (SSE-KMS) o utilizza le chiavi di crittografia gestite di Amazon S3 (SSE-S3). Per ulteriori informazioni, consulta Protezione dei dati con la crittografia lato server.	Sì
`x-amz-checksum-crc32` , `x-amz-checksum-crc32c` , `x-amz-checksum-sha1` , `x-amz-checksum-sha256`	Intestazioni che contengono il checksum o il digest dell'oggetto. Viene impostata al massimo una intestazione alla volta, a seconda dell'algoritmo di checksum che Amazon S3 deve utilizzare. Per ulteriori informazioni sulla scelta dell'algoritmo di checksum, consulta Verifica dell'integrità degli oggetti.	No
`x-amz-version-id`	Versione dell'oggetto. Quando abiliti il controllo delle versioni in un bucket, Amazon S3 assegna un ID versione agli oggetti aggiunti al bucket. Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.	No
`x-amz-delete-marker`	Contrassegno booleano che indica se l'oggetto è un contrassegno di eliminazione. Questo contrassegno viene utilizzato solo nei bucket in cui è abilitato il controllo delle versioni.	No
`x-amz-storage-class`	Classe di archiviazione utilizzata per l'archiviazione dell'oggetto. Per ulteriori informazioni, consulta Utilizzo delle classi di storage di Amazon S3.	Sì

Nome	Descrizione	L'utente può modificare il valore?
`x-amz-website-redirect-location`	Intestazione che reindirizza le richieste per l'oggetto associato a un altro oggetto nello stesso bucket o a un URL esterno. Per ulteriori informazioni, consulta (Facoltativo) Configurazione del reindirizzamento di una pagina Web.	Sì
`x-amz-server-side-encryption-aws-kms-key-id`	Un'intestazione che indica l'ID della chiave KMS di crittografia AWS KMS simmetrica utilizzata per crittografare l'oggetto. Questa intestazione viene utilizzata solo quando è presente l'intestazione `x-amz-server-side-encryption` e ha il valore `aws:kms`.	Sì
`x-amz-server-side-encryption-customer-algorithm`	Intestazione che indica se è abilitata la crittografia lato server con le chiavi di crittografia fornite dal cliente (SSE-C). Per ulteriori informazioni, consulta Utilizzo della crittografia lato server con chiavi fornite dal cliente (SSE-C).	Sì
`x-amz-tagging`	Il set di tag per l'oggetto. Il set di tag deve essere codificato sotto forma di parametri della URL Query.	Sì

Metadati di oggetti definiti dall'utente

Quando si carica un oggetto, è anche possibile assegnare metadata a esso. Queste informazioni facoltative vengono fornite come coppia nome-valore (chiave-valore) quando si invia una richiesta `PUT` o `POST` per creare l'oggetto. Quando si caricano gli oggetti utilizzando la REST API, i nomi facoltativi dei metadati definiti dall'utente devono iniziare con `x-amz-meta-` per distinguerli dalle altre intestazioni HTTP. Quando si recupera l'oggetto utilizzando REST API, questo prefisso viene restituito. Quando si caricano gli oggetti utilizzando l'API SOAP, il prefisso non è necessario. Quando si recupera l'oggetto mediante l'API SOAP, il prefisso viene rimosso, indipendentemente dall'API utilizzata per caricare l'oggetto.

> **Note**
>
> Il supporto di SOAP su HTTP non viene più utilizzato ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non sono supportate per SOAP. Invece di usare SOAP, ti consigliamo di utilizzare l'API REST o gli SDK. AWS

Quando i metadati vengono recuperati tramite REST API, Amazon S3 riunisce le intestazioni con lo stesso nome (senza distinzione tra maiuscole e minuscole) in un elenco delimitato da virgole. I metadata contenenti caratteri non stampabili non vengono restituiti. Al contrario, viene restituita l'intestazione `x-amz-missing-meta` con il numero di voci di metadata non stampabili come valore. L'operazione `HeadObject` richiama i metadati da un oggetto senza restituire l'oggetto stesso. Questa operazione è utile se sei interessato solo ai metadati di un oggetto. Per utilizzare `HEAD` è necessario disporre dell'accesso `READ` all'oggetto. Per ulteriori informazioni, consulta il riferimento HeadObjectall'API di Amazon Simple Storage Service.

I metadata definiti dall'utente sono un set di coppie chiave-valore. Amazon S3 archivia le chiavi dei metadati definiti dall'utente in caratteri minuscoli.

Amazon S3 consente caratteri Unicode arbitrari nei valori dei metadati.

Per evitare problemi relativi alla presentazione di questi valori di metadati, è necessario conformarsi all'utilizzo dei caratteri US-ASCII quando si usano REST e UTF-8 con SOAP o caricamenti basati su browser tramite `POST`.

Quando si utilizzano caratteri non US-ASCII nei valori dei metadati, la stringa Unicode fornita viene esaminata per i caratteri non US-ASCII. I caratteri dei valori di tali header sono decodificati come da RFC 2047 prima di memorizzarli e codificarli come da RFC 2047 per renderli sicuri per la posta elettronica prima di restituirli. Se la stringa contiene solo caratteri US-ASCII, viene presentata così com'è.

Di seguito è riportato un esempio.

```
PUT /Key HTTP/1.1
Host: DOC-EXAMPLE-BUCKET1.s3.amazonaws.com
x-amz-meta-nonascii: ÄMÄZÕÑ S3

HEAD /Key HTTP/1.1
```

```
Host: DOC-EXAMPLE-BUCKET1.s3.amazonaws.com
x-amz-meta-nonascii: =?UTF-8?B?w4PChE3Dg8KEWsODwpXDg8KRIFMz?=

PUT /Key HTTP/1.1
Host: DOC-EXAMPLE-BUCKET1.s3.amazonaws.com
x-amz-meta-ascii: AMAZONS3

HEAD /Key HTTP/1.1
Host: DOC-EXAMPLE-BUCKET1.s3.amazonaws.com
x-amz-meta-ascii: AMAZONS3
```

> **Note**
>
> L'intestazione della richiesta PUT è limitata a una dimensione di 8 KB. Nell'intestazione della richiesta PUT, la dimensione dei metadata definiti dall'utente è limitata a 2 KB. La dimensione dei metadata definiti dall'utente viene calcolata sommando il numero di byte della codifica UTF-8 di ogni chiave e valore.

Per informazioni sulla modifica dei metadati dell'oggetto dopo il caricamento mediante la creazione di una copia dell'oggetto, la modifica e la sostituzione dell'oggetto precedente o la creazione di una nuova versione, consulta Modifica dei metadati degli oggetti nella console di Amazon S3.

Modifica dei metadati degli oggetti nella console di Amazon S3

Puoi utilizzare la console di Amazon S3 per modificare i metadati degli oggetti S3 esistenti. Alcuni metadati vengono impostati da Amazon S3 quando carichi l'oggetto. Ad esempio, `Content-Length` e `Last-Modified` sono campi di metadati degli oggetti definiti dal sistema che non possono essere modificati da un utente.

Puoi anche impostare alcuni metadati quando carichi l'oggetto oppure aggiungerli in un secondo momento a seconda delle necessità. Ad esempio, è possibile che tu disponga di un insieme di oggetti che inizialmente hai memorizzato nella classe di storage `STANDARD`. Nel corso del tempo, potrebbe non essere più necessario che questi dati siano altamente disponibili. Quindi modifichi la classe di storage in `GLACIER` modificando il valore della chiave `x-amz-storage-class` da `STANDARD` a `GLACIER`.

Note

Prendi in considerazione i seguenti problemi quando modifichi i metadati degli oggetti in Amazon S3:

- Questa operazione crea una copia dell'oggetto con le impostazioni aggiornate e la data dell'ultima modifica. Se è abilitata la funzione Controllo delle versioni S3, viene creata una nuova versione dell'oggetto e l'oggetto esistente diventa una versione precedente. Se il controllo delle versioni S3 non è abilitato, una nuova copia dell'oggetto sostituisce l'oggetto originale. Il ruolo Account AWS associato al ruolo IAM che modifica la proprietà diventa anche il proprietario del nuovo oggetto o (versione dell'oggetto).
- Per utilizzare la console Amazon S3 per modificare i metadati di un oggetto con tag definiti dall'utente, devi disporre anche dell'autorizzazione. `s3:GetObjectTagging` Se utilizzi la console Amazon S3 per modificare i metadati di un oggetto che non ha tag definiti dall'utente ma ha una dimensione superiore a 16 MB, devi disporre anche dell'autorizzazione. `s3:GetObjectTagging`

  Se la policy del bucket di destinazione nega l'`s3:GetObjectTagging`azione, i metadati dell'oggetto verranno aggiornati, ma i tag definiti dall'utente verranno rimossi dall'oggetto e riceverai un errore.
- La modifica dei metadati aggiorna i valori per i nomi chiave esistenti.
- Gli oggetti crittografati con chiavi di crittografia fornite dal cliente (SSE-C) non possono essere copiati utilizzando la console. È necessario utilizzare l' AWS CLI AWS SDK o l'API REST di Amazon S3.

Warning

Quando si modificano i metadati delle cartelle, attendi il completamento dell'operazione `Edit metadata` prima di aggiungere nuovi oggetti alla cartella. In caso contrario, potrebbero essere modificati anche i nuovi oggetti.

Negli argomenti seguenti viene descritto come modificare i metadati di un oggetto utilizzando la console di Amazon S3.

Aggiunta di metadati definiti dal sistema

È possibile configurare alcuni ma non tutti i metadati di sistema per un oggetto di S3. Per un elenco dei metadati definiti dal sistema e per sapere se è possibile modificarne i valori, consulta Metadata di oggetti definiti dal sistema.

Per modificare i metadati definiti dal sistema di un oggetto

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Accedi al bucket o alla cartella Amazon S3 e seleziona la casella di controllo a sinistra dei nomi degli oggetti con i metadati da modificare.
3. Dal menu Operazioni, scegli Operazioni di modifica e quindi Modifica metadati.
4. Esamina gli oggetti elencati e scegli Aggiungi metadati.
5. Per Type (Tipo) di metadati, selezionare System-defined (Definiti dal sistema).
6. Specificare una Key (Chiave) univoca e il Value (Valore)dei metadati.
7. Per modificare metadati aggiuntivi, scegliere Add metadata (Aggiungi metadati). Puoi anche scegliere Rimuovi per rimuovere un set di type-key-values.
8. Una volta terminato, scegli Modifica metadati e Amazon S3 modificherà i metadati degli oggetti specificati.

Aggiunta di metadati definiti dall'utente

È possibile modificare i metadati definiti dall'utente di un oggetto combinando il prefisso dei metadati `x-amz-meta-` e un nome scelto per creare una chiave personalizzata. Ad esempio, se si aggiunge il nome personalizzato `alt-name`, la chiave dei metadati sarà `x-amz-meta-alt-name`.

I metadati definiti dall'utente possono avere una dimensione massima di 2 KB. Per calcolare la dimensione totale dei metadati definiti dall'utente, somma il numero di byte nella codifica UTF-8 per ogni chiave e valore. Sia le chiavi che i relativi valori devono essere conformi agli standard US-ASCII. Per ulteriori informazioni, consulta Metadati di oggetti definiti dall'utente.

Per modificare i metadati definiti dall'utente di un oggetto

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

2. Nell'elenco Bucket, scegli il nome del bucket che contiene gli oggetti a cui desideri aggiungere metadati.

   Se necessario, puoi passare a una cartella.
3. Nell'elenco Oggetti seleziona la casella di controllo accanto ai nomi degli oggetti a cui desideri aggiungere metadati.
4. Dal menu Operazioni, scegli Modifica metadati.
5. Esamina gli oggetti elencati e scegli Aggiungi metadati.
6. Per Tipo, scegli Definito dall'utente.
7. Immetti una Chiave univoca e personalizzata dopo `x-amz-meta-`. Immettere anche un Value (Valore) metadati.
8. Per aggiungere metadati, scegliere Add metadata (Aggiungi metadati). Puoi anche scegliere Rimuovi per rimuovere un set di type-key-values.
9. Seleziona Modifica metadati.

   Amazon S3 modifica i metadati degli oggetti specificati.

Caricamento degli oggetti

Quando un file viene caricato in Amazon S3, viene archiviato come oggetto S3. Gli oggetti sono composti dai dati e dai metadata dei file che descrivono l'oggetto. Un bucket può avere un numero illimitato di oggetti. Per caricare file e cartelle in un bucket Amazon S3, è necessario disporre delle autorizzazioni in scrittura per il bucket. Per ulteriori informazioni sulle autorizzazioni di accesso, consultare Identity and Access Management per Amazon S3.

In un bucket S3 è possibile caricare qualsiasi tipo di file: immagini, backup, dati, film e così via. La dimensione massima di un file che è possibile caricare utilizzando la console di Amazon S3 è 160 GB. Per caricare un file di dimensioni superiori a 160 GB, usa AWS Command Line Interface (AWS CLI), gli AWS SDK o l'API REST di Amazon S3.

Se si carica un oggetto con un nome della chiave già esistente in un bucket che supporta la funzione Controllo delle versioni, Amazon S3 crea un'altra versione dell'oggetto anziché sostituire l'oggetto esistente. Per ulteriori informazioni sulla funzione Controllo delle versioni, consulta Utilizzo della console S3.

A seconda della dimensione dei dati da caricare, in Amazon S3 sono disponibili le seguenti opzioni:

- Carica un oggetto con un'unica operazione utilizzando gli AWS SDK, l'API REST oppure AWS CLI — Con una singola PUT operazione, puoi caricare un singolo oggetto di dimensioni fino a 5 GB.
- Carica un singolo oggetto tramite la console di Amazon S3: con la console di Amazon S3, è possibile caricare un singolo oggetto fino a 160 GB di dimensioni.
- Carica un oggetto in più parti utilizzando gli AWS SDK, l'API REST oppure AWS CLI: utilizzando l'operazione API di caricamento multiparte, puoi caricare un singolo oggetto di grandi dimensioni, di dimensioni fino a 5 TB.

  L'operazione API per il caricamento in più parti è concepita per migliorare l'esperienza di caricamento per gli oggetti di dimensioni maggiori. È possibile caricare un oggetto in parti. Queste parti possono essere caricate in modo indipendente, in qualsiasi ordine e in parallelo. È possibile utilizzare un caricamento in più parti per gli oggetti con una dimensione compresa tra 5 MB e 5 TB. Per ulteriori informazioni, consulta Caricamento e copia di oggetti utilizzando il caricamento in più parti.

Al momento del caricamento, l'oggetto viene crittografato automaticamente per impostazione predefinita utilizzando la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3). Quando lo scarichi, l'oggetto viene decrittato. Per ulteriori informazioni, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3 e Protezione dei dati con la crittografia.

Quando carichi un oggetto, se desideri utilizzare un diverso tipo di crittografia predefinita, puoi anche specificare la crittografia lato server con le chiavi AWS Key Management Service (AWS KMS) (SSE-KMS) nelle tue richieste PUT S3 o impostare la configurazione di crittografia predefinita nel bucket di destinazione per utilizzare SSE-KMS per crittografare i dati. Per ulteriori informazioni su SSE-KMS, consulta Specifica della crittografia lato server con AWS KMS (SSE-KMS). Se desideri utilizzare una chiave KMS di proprietà di un account diverso, devi avere l'autorizzazione necessaria per l'uso della chiave. Per ulteriori informazioni sulle autorizzazioni tra account per le chiavi KMS, vedi Creazione di chiavi KMS utilizzabili da altri account nella Guida per gli sviluppatori di AWS Key Management Service .

Se riscontri un errore di accesso negato (403 proibito) in Amazon S3, Risoluzione dei problemi relativi agli errori di accesso negato (403 Accesso negato) in Amazon S3 consulta la pagina per saperne di più sulle cause più comuni.

Utilizzo della console S3

Questa procedura spiega come caricare oggetti e cartelle in un bucket Amazon S3 utilizzando la console.

Quando carichi un oggetto, il nome della chiave oggetto è costituito dal nome del file e da qualsiasi prefisso facoltativo. Nella console di Amazon S3, puoi creare cartelle per organizzare i tuoi oggetti. In Amazon S3, le cartelle sono rappresentate come prefissi visualizzati nel nome della chiave oggetto. Se carichi un singolo oggetto in una cartella nella console di Amazon S3, il nome della cartella viene incluso nel nome della chiave oggetto.

Ad esempio, se carichi un oggetto denominato `sample1.jpg` in una cartella denominata `backup`, il nome della chiave è `backup/sample1.jpg`. Tuttavia, l'oggetto viene visualizzato nella console come `sample1.jpg` nella cartella `backup`. Per ulteriori informazioni sui nomi delle chiavi, consultare Utilizzo dei metadati degli oggetti.

> **Note**
>
> Se rinomini un oggetto o modifichi una delle proprietà nella console di Amazon S3, ad esempio Classe di storage, Crittografia o Metadati, viene creato un nuovo oggetto per sostituire quello precedente. Se è abilitata la funzione Controllo delle versioni S3, viene creata una nuova versione dell'oggetto e l'oggetto esistente diventa una versione precedente. Il ruolo che modifica la proprietà diventa anche il proprietario del nuovo oggetto o della versione dell'oggetto.

Quando si carica una cartella, Amazon S3 carica nel bucket tutti i file e le sottocartelle inclusi nella cartella specificata, quindi assegna un nome della chiave dell'oggetto, ossia una combinazione del nome del file caricato e del nome della cartella. Ad esempio, se si carica una cartella denominata `/images` contenente due file, `sample1.jpg` e `sample2.jpg`, Amazon S3 carica i file, quindi assegna i nomi delle chiavi corrispondenti, `images/sample1.jpg` e `images/sample2.jpg`. I nomi delle chiavi includono il nome della cartella come prefisso. La console di Amazon S3 visualizza solo la parte del nome della chiave che segue l'ultimo simbolo `/`. Ad esempio, in una cartella di `images`, gli oggetti `images/sample1.jpg` e `images/sample2.jpg` sono visualizzati come `sample1.jpg` e `sample2.jpg`.

Per caricare cartelle e file in un bucket S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckets (Bucket) scegliere il nome del bucket in cui si desidera caricare le cartelle o i file.
4. Scegli Carica.
5. Nella finestra Carica completa una delle seguenti operazioni:
   - Trascina e rilascia file e cartelle nella finestra Carica .
   - Scegli Aggiungi file o Aggiungi cartella, seleziona i file o le cartelle da caricare e scegli Apri.
6. Per abilitare il controllo delle versioni, in Destinazione, seleziona Abilita controllo delle versioni del bucket.
7. Per caricare i file e le cartelle elencati senza configurare ulteriori opzioni di caricamento, nella parte inferiore della pagina scegli Carica.

   Amazon S3 caricherà i tuoi oggetti e le tue cartelle. Al termine del caricamento viene visualizzato un messaggio di esito positivo nella pagina Carica: stato.

Per configurare proprietà aggiuntive dell'oggetto

1. Per modificare le autorizzazioni della lista di controllo degli accessi, scegli Permissions (Autorizzazioni).
2. In Access control list (ACL) Lista di controllo degli accessi (ACL), modifica le autorizzazioni.

   Per informazioni sulle autorizzazioni di accesso agli oggetti, consulta Utilizzo della console S3 per impostare le autorizzazioni ACL per un oggetto. Puoi concedere l'accesso in lettura ai tuoi oggetti al pubblico (chiunque) per tutti i file che stai caricando. Ti consigliamo di non modificare l'impostazione di default per l'accesso pubblico in lettura. La concessione dell'accesso pubblico in lettura si applica a un piccolo sottoinsieme di casi d'uso, ad esempio quando i bucket vengono usati per i siti Web. È sempre possibile apportare modifiche alle autorizzazioni dell'oggetto dopo averlo caricato.
3. Per configurare altre proprietà scegli Properties (Proprietà).
4. Nella sezione Classe di storage seleziona la classe di storage per i file che si stanno caricando.

Per ulteriori informazioni sulle classi di storage, consulta Utilizzo delle classi di storage di Amazon S3.

5. Per aggiornare le impostazioni di crittografia per gli oggetti, in Impostazioni di crittografia lato server completa le operazioni riportate di seguito.

   a. Scegli Specify an encryption key (Specifica una chiave di crittografia).

   b. In Impostazioni di crittografia, scegli Utilizza le impostazioni del bucket per la crittografia predefinita o Ignora le impostazioni del bucket per la crittografia predefinita.

   c. Se scegli Ignora le impostazioni del bucket per la crittografia predefinita, dovrai configurare le seguenti impostazioni di crittografia.

      - Per crittografare i file caricati utilizzando chiavi gestite da Amazon S3, seleziona Chiave gestita da Amazon S3 (SSE-S3).

        Per ulteriori informazioni, consulta Uso della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).

      - Per crittografare i file caricati utilizzando le chiavi memorizzate in AWS Key Management Service (AWS KMS), scegli AWS Key Management Service key (SSE-KMS). Quindi scegli una delle seguenti opzioni per Chiave AWS KMS :

        - Per scegliere da un elenco di chiavi KMS disponibili, seleziona Scegli tra le chiavi AWS KMS keys, quindi scegli la chiave KMS dall'elenco delle chiavi disponibili.

          In questo elenco vengono visualizzate sia la chiave Chiave gestita da AWS (`aws/s3`) che quella gestita dal cliente. Per ulteriori informazioni sulle chiavi gestite dal cliente, consulta Chiavi gestite dal cliente e chiavi AWS nella Guida per gli sviluppatori di AWS Key Management Service .

        - Per inserire l'ARN della chiave KMS, scegli Inserisci AWS KMS key ARN, quindi inserisci l'ARN della chiave KMS nel campo visualizzato.

        - Per creare una nuova chiave gestita dal cliente nella AWS KMS console, scegli Crea una chiave KMS.

          Per ulteriori informazioni sulla creazione di una AWS KMS key, consulta Creating keys nella AWS Key Management Service Developer Guide.

> **Important**
>
> Puoi utilizzare solo le chiavi KMS disponibili nello Regione AWS stesso bucket. La console Amazon S3 elenca solo le prime 100 chiavi KMS nella stessa regione del bucket. Per utilizzare una chiave KMS non elencata, devi inserire l'ARN della chiave KMS. Se desideri utilizzare una chiave KMS di proprietà di un account diverso, è necessario innanzitutto disporre dell'autorizzazione necessaria per l'uso della chiave e quindi inserire l'ARN della chiave KMS.
> Amazon S3 supporta solo chiavi KMS di crittografia simmetriche e non chiavi KMS asimmetriche. Per ulteriori informazioni, consulta Identificazione delle chiavi KMS simmetriche e asimmetriche nella Guida per gli sviluppatori di AWS Key Management Service .

6. Per utilizzare checksum aggiuntivi, scegli On (Attivato). Per Checksum function (Funzione checksum), scegli la funzione che desideri utilizzare. Amazon S3 calcola e archivia il valore del checksum dopo aver ricevuto l'intero oggetto. Puoi utilizzare la casella Precalculated value (Valore precalcolato) per fornire un valore precalcolato. In tal caso, Amazon S3 confronta il valore specificato con il valore calcolato. Se i due valori non corrispondono, Amazon S3 genera un errore.

   I checksum aggiuntivi ti consentono di specificare l'algoritmo di checksum che desideri utilizzare per verificare i dati. Per ulteriori informazioni sui checksum aggiuntivi, consulta Verifica dell'integrità degli oggetti.

7. Per aggiungere tag a tutti gli oggetti che si stanno caricando, scegliere Add tag (Aggiungi tag). Immetti un nome di tag nel campo Chiave. Immetti un valore per il tag.

   Il tagging ti consente di catalogare lo storage. Ogni tag è una coppia chiave-valore. I valori delle chiavi e dei tag fanno distinzione tra maiuscole e minuscole. Puoi avere un massimo di 10 tag per oggetto. Una chiave di tag può essere composta da un massimo di 128 caratteri Unicode e i valori di tag possono essere composti da un massimo di 255 caratteri Unicode. Per ulteriori informazioni sui tag degli oggetti, consulta Suddivisione in categorie dello storage utilizzando i tag.

8. Per aggiungere metadati, seleziona Aggiungi metadati.

   a. In Tipo seleziona Definito dal sistema o Definito dall'utente.

Per i metadati definiti dal sistema, puoi selezionare le intestazioni HTTP comuni, ad esempio Content-Type e Content-Disposition. Per un elenco di metadati definiti dal sistema e informazioni sulla possibilità di aggiungere il valore, consulta Metadata di oggetti definiti dal sistema. Eventuali metadata che iniziano con il prefisso `x-amz-meta-` sono considerati come metadata definiti dall'utente. I metadata definiti dall'utente vengono archiviati con l'oggetto e vengono restituiti quando si scarica l'oggetto. Sia le chiavi che i relativi valori devono essere conformi agli standard US-ASCII. I metadata definiti dall'utente possono avere una dimensione massima di 2 KB. Per ulteriori informazioni sui metadati definiti dal sistema e definiti dall'utente, consulta Utilizzo dei metadati degli oggetti.

b. Per Chiave, seleziona una chiave.

c. Digitare un valore per la chiave.

9. Per caricare i tuoi oggetti, scegli Carica.

   Amazon S3 caricherà l'oggetto. Al termine del caricamento, sarà visualizzato un messaggio di successo nella pagina Carica: stato .

10. Scegliere Exit (Esci).

Utilizzo degli SDK AWS

Puoi utilizzare gli AWS SDK per caricare oggetti in Amazon S3. Gli SDK offrono le librerie wrapper per facilitare il caricamento dei dati. Per informazioni, consulta l'elenco degli SDK supportati.

Di seguito sono riportati degli esempi con alcuni SDK selezionati:

.NET

Il seguente esempio di codice #C crea due oggetti con due richieste `PutObjectRequest`:

- La prima richiesta `PutObjectRequest` salva una stringa di testo come dati dell'oggetto di esempio. Specifica inoltre il nome del bucket e il nome della chiave dell'oggetto.
- La seconda richiesta `PutObjectRequest` carica un file specificando il nome file. Questa richiesta specifica anche l'intestazione `ContentType` e i metadati opzionali dell'oggetto (un titolo).

Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadObjectTest
    {
        private const string bucketName = "*** bucket name ***";
        // For simplicity the example creates two objects from the same file.
        // You specify key names for these objects.
        private const string keyName1 = "*** key name for first object created ***";
        private const string keyName2 = "*** key name for second object created
 ***";
        private const string filePath = @"*** file path ***";
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.EUWest1;

        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            WritingAnObjectAsync().Wait();
        }

        static async Task WritingAnObjectAsync()
        {
            try
            {
                // 1. Put object-specify only key name for the new object.
                var putRequest1 = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName1,
                    ContentBody = "sample text"
                };

                PutObjectResponse response1 = await
 client.PutObjectAsync(putRequest1);
```

```
                // 2. Put the object-set ContentType and add metadata.
                var putRequest2 = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName2,
                    FilePath = filePath,
                    ContentType = "text/plain"
                };

                putRequest2.Metadata.Add("x-amz-meta-title", "someTitle");
                PutObjectResponse response2 = await
 client.PutObjectAsync(putRequest2);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine(
                        "Error encountered ***. Message:'{0}' when writing an
 object"
                        , e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine(
                    "Unknown encountered on server. Message:'{0}' when writing an
 object"
                    , e.Message);
            }
        }
    }
}
```

Java

L'esempio seguente crea due oggetti. Il primo ha una stringa di testo come dati e il secondo è un file. L'esempio crea il primo oggetto specificando il nome del bucket, la chiave dell'oggetto e i dati di testo direttamente in una chiamata a `AmazonS3Client.putObject()`. L'esempio crea il secondo oggetto utilizzando una richiesta `PutObjectRequest` che specifica il nome del bucket, la chiave dell'oggetto e il percorso del file. La richiesta `PutObjectRequest` specifica anche l'intestazione `ContentType` e i metadati del titolo.

Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
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

Nell'esempio di seguito viene caricato un file esistente in un bucket Amazon S3 in una regione specifica.

```
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

Questo esempio ti guida nell'utilizzo delle classi di AWS SDK for PHP per caricare un oggetto di dimensioni fino a 5 GB. Per i file di dimensioni maggiori è necessario utilizzare l'operazione API per il caricamento in più parti. Per ulteriori informazioni, consulta Caricamento e copia di oggetti utilizzando il caricamento in più parti.

Per ulteriori informazioni sull'API AWS SDK for Ruby, AWS vai a SDK for Ruby - Versione 2.

Example – Creazione di un oggetto in un bucket Amazon S3 tramite il caricamento dei dati

Nel seguente esempio di codice PHP viene creato un oggetto in un bucket specificato mediante il caricamento dei dati con il metodo `putObject()`.

```
 require 'vendor/autoload.php';

use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

try {
    // Upload data.
    $result = $s3->putObject([
        'Bucket' => $bucket,
        'Key'    => $keyname,
        'Body'   => 'Hello, world!',
        'ACL'    => 'public-read'
    ]);

    // Print the URL to the object.
    echo $result['ObjectURL'] . PHP_EOL;
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Ruby

La AWS SDK for Ruby - Versione 3 offre due modi per caricare un oggetto su Amazon S3. Il primo metodo consiste nell'utilizzare un uploader di file gestito che facilita il caricamento dei file di qualsiasi dimensione dal disco. Per utilizzare tale metodo:

1. Crea un'istanza della classe `Aws::S3::Resource`.

2. Fare riferimento all'oggetto di destinazione in base al nome del bucket e alla chiave. Gli oggetti sono in un bucket e dispongono di chiavi univoche con le quali vengono identificati.
3. Eseguire la chiamata `#upload_file` sull'oggetto.

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__
```

Il secondo modo in cui AWS SDK for Ruby - Version 3 può caricare un oggetto utilizza il `#put` metodo di. `Aws::S3::Object` Questo metodo è utile se l'oggetto è una stringa o un oggetto I/O che non è un file su disco. Per utilizzare tale metodo:

1. Crea un'istanza della classe `Aws::S3::Resource`.
2. Fare riferimento all'oggetto di destinazione in base al nome del bucket e alla chiave.
3. Eseguire la chiamata `#put` passando la stringa o l'oggetto 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__
```

Utilizzo di REST API

Per caricare un oggetto puoi inviare richieste REST. È possibile inviare una richiesta PUT per caricare i dati in una singola operazione. Per ulteriori informazioni, consulta PUT Object.

Usando il AWS CLI

È possibile inviare una richiesta PUT per caricare un oggetto di un massimo di 5 GB in una singola operazione. Per ulteriori informazioni ed esempi, consulta l'esempio `PutObject` in Riferimento ai comandi della AWS CLI .

Caricamento e copia di oggetti utilizzando il caricamento in più parti

Il caricamento in più parti consente di caricare un singolo oggetto come un insieme di parti. Ciascuna parte è una parte contigua dei dati dell'oggetto. È possibile caricare queste parti dell'oggetto in modo indipendente e in qualsiasi ordine. Se la trasmissione di una parte non riesce, è possibile ritrasmettere tale parte senza influire sulle altre. Una volta caricate tutte le parti dell'oggetto, Amazon S3 le assembla e crea l'oggetto. In generale, quando la dimensione dell'oggetto raggiunge i 100 MB, si consiglia di valutare la possibilità di eseguire caricamenti in più parti anziché caricare l'oggetto in una singola operazione.

Il caricamento in più parti comporta i vantaggi riportati di seguito.

- Velocità effettiva migliorata: puoi caricare le parti in parallelo per migliorare la velocità effettiva.
- Ripristino rapido dai problemi di rete: la dimensione più piccola delle parti riduce al minimo l'impatto del riavvio di un caricamento fallito a causa di un errore di rete.

- Messa in pausa e ripresa dei caricamenti dell'oggetto: puoi caricare le parti dell'oggetto nel corso del tempo. Una volta avviato, un caricamento in più parti continua finché non viene completato o interrotto in modo esplicito.
- Avvio di un caricamento prima di conoscere la dimensione finale dell'oggetto: puoi caricare un oggetto mentre viene creato.

È consigliabile utilizzare il caricamento in più parti come indicato di seguito:

- Se si stanno caricando oggetti di grandi dimensioni in una rete a banda larga stabile, utilizzare il caricamento in più parti per ottimizzare l'uso della larghezza di banda disponibile caricando le parti degli oggetti in parallelo per garantire prestazioni ottimali in più thread.
- Se il caricamento viene eseguito su una rete non stabile, utilizzare il caricamento in più parti per aumentare la resilienza agli errori di rete evitando di riavviare più volte il caricamento. Quando si utilizza il caricamento in più parti, è necessario ritentare il caricamento solo delle parti interrotte durante il caricamento stesso. Non è necessario riavviare il caricamento dell'oggetto dall'inizio.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory. Per ulteriori informazioni sull'utilizzo del caricamento in più parti con S3 Express One Zone e i bucket di directory, consulta Utilizzo di caricamenti multiparte con bucket di directory.

Processo di caricamento in più parti

Il caricamento in più parti è un processo in tre fasi: avvio del caricamento, caricamento delle parti dell'oggetto e, una volta completato il caricamento di tutte le parti, completamento del caricamento in più parti. Quando riceve la conferma di completamento del caricamento in più parti, Amazon S3 crea l'oggetto dalle singole parti caricate. È quindi possibile accedere all'oggetto così come avviene con qualsiasi altro oggetto nel bucket.

È possibile elencare tutti i caricamenti in più parti in corso oppure ottenere un elenco delle parti caricate per un caricamento in più parti specifico. Ognuna di queste operazioni viene descritta in questa sezione.

Avvio del caricamento in più parti

Quando invii una richiesta di avvio di un caricamento in più parti, Amazon S3 restituisce una risposta con un ID di caricamento, che è un identificativo univoco per il caricamento in più parti. È necessario includere questo ID di caricamento ogni volta che si caricano o si elencano le parti oppure ogni volta che si completa o si interrompe un caricamento. Se si desidera fornire metadata che descrivano l'oggetto in fase di caricamento, è necessario specificarli nella richiesta di avvio del caricamento in più parti.

Caricamento delle parti

Quando si carica una parte, oltre all'ID di caricamento è necessario specificare il numero della parte. È possibile scegliere qualsiasi numero compreso tra 1 e 10.000. Il numero della parte identifica in modo univoco una parte e la relativa posizione nell'oggetto che si sta caricando. Il numero della parte scelto non deve essere in sequenza (ad esempio può essere 1, 5 e 14). Se si carica una nuova parte che utilizza lo stesso numero di una parte caricata in precedenza, quest'ultima viene sovrascritta.

Quando carichi una parte, Amazon S3 restituisce un tag di entità (ETag) per la parte come intestazione nella risposta. Per ogni caricamento di parte è necessario registrare il numero della parte e il valore ETag. Occorre includere questi valori nella successiva richiesta di complemento del caricamento in più parti. Ogni parte avrà il proprio eTag al momento del caricamento. Tuttavia, una volta completato il caricamento in più parti e tutte le parti consolidate, tutte le parti saranno riunite in un unico ETag come checksum di checksum.

> **Note**
>
> Dopo aver avviato un caricamento in più parti e aver caricato una o più parti, è necessario completare o interrompere questa operazione per interrompere l'addebito per l'archiviazione delle parti caricate. Solo dopo aver completato o interrotto un caricamento in più parti, Amazon S3 libererà spazio di storage per le parti e interromperà l'addebito per lo storage delle parti.
>
> Una volta interrotto, non è possibile caricare di nuovo una parte che utilizza tale ID di caricamento. Se eventuali caricamenti di parti erano in corso, possono essere eseguiti correttamente o meno anche dopo l'interruzione del caricamento. Per liberare completamente lo spazio di archiviazione utilizzato da tutte le parti, è necessario interrompere un caricamento in più parti solo al termine di tutti i caricamenti delle parti.

Completamento del caricamento in più parti

Una volta completato un caricamento in più parti, Amazon S3 crea un oggetto concatenando le parti in ordine crescente in base al numero della parte. Se nella richiesta di avvio del caricamento in più parti sono stati forniti i metadati dell'oggetto, Amazon S3 li associa all'oggetto. Una volta completata la richiesta, le parti non esisteranno più.

La richiesta di completamento del caricamento in più parti deve includere l'ID di caricamento e un elenco sia dei numeri delle parti sia dei valori ETag corrispondenti. La risposta di Amazon S3 include un ETag che identifica in modo univoco i dati oggetto combinati. Questo ETag non è necessariamente un hash MD5 dei dati dell'oggetto.

Chiamate di caricamento in più parti di esempio

Per questo esempio, si supponga di generare un caricamento in più parti di un file da 100 GB. In questo caso, sarebbero disponibili le seguenti chiamate API per l'intero processo. Ci sarebbero un totale di 1002 chiamate API.

- Una chiamata `CreateMultipartUpload` per avviare il processo.
- 1000 chiamate `UploadPart` singole, ognuna delle quali carica una parte di 100 MB, per una dimensione totale di 100 GB.
- Una chiamata `CompleteMultipartUpload` per terminare il processo.

Elenchi dei caricamenti in più parti

È possibile elencare le parti di un caricamento in più parti specifico o tutti i caricamenti in più parti in corso. L'operazione di creazione dell'elenco delle parti restituisce informazioni sulle parti coinvolte in un caricamento in più parti specifico. Per ogni richiesta di elenco delle parti, Amazon S3 restituisce informazioni sulle parti per il caricamento in più parti specificato, fino a un massimo di 1000 parti. Se nel caricamento sono presenti più di 1000 parti, è necessario inviare una serie di richieste di elenco delle parti per recuperare tutte le parti. Tieni presente che l'elenco di parti restituito non include le parti per le quali non è stato completato il caricamento. L'operazione list multipart uploads (elenco dei caricamenti in più parti) consente di ottenere l'elenco dei caricamenti in più parti in corso.

Un caricamento in più parti in corso è un caricamento avviato, ma non ancora completato o annullato. Ogni richiesta restituisce al massimo 1.000 caricamenti in più parti. Se sono in corso più di 1.000 caricamenti in più parti, è necessario inviare richieste aggiuntive per recuperare i caricamenti rimanenti. Utilizza l'elenco restituito solo per la verifica. Non utilizzarlo per inviare la richiesta complete multipart upload (completamento del caricamento in più parti). Al contrario, mantieni

il tuo elenco dei numeri delle parti specificato durante il caricamento delle parti e i valori ETag corrispondenti restituiti da Amazon S3.

Checksum con operazioni di caricamento in più parti

Quando carichi un oggetto in Amazon S3 puoi specificare un algoritmo di checksum che Amazon S3 deve utilizzare. Per verificare l'integrità dei dati Amazon S3 utilizza MD5 per impostazione di default, tuttavia puoi specificare di usare un algoritmo di checksum aggiuntivo. Quando utilizza MD5, Amazon S3 calcola il checksum dell'intero oggetto in più parti dopo il completamento del caricamento. Questo checksum non è un checksum dell'intero oggetto, ma il checksum dei checksum di ogni singola parte.

Quando chiedi ad Amazon S3 di utilizzare checksum aggiuntivi, Amazon S3 calcola il valore del checksum per ogni parte e archivia i valori. È possibile utilizzare l'API o l'SDK per recuperare il valore di checksum per le singole parti usando `GetObject` o `HeadObject`. Se desideri recuperare i valori di checksum per singole parti di caricamenti in più parti ancora in corso, puoi utilizzare `ListParts`.

> **Important**
>
> Se utilizzi un caricamento in più parti con checksum aggiuntivi, i numeri delle parti in più parti devono essere consecutivi. Quando usi i checksum aggiuntivi, se tenti di completare una richiesta di caricamento in più parti con numeri parte non consecutivi, Amazon S3 genera un errore HTTP `500 Internal Server Error`.

Per ulteriori informazioni sul funzionamento dei checksum con oggetti in più parti, consulta Verifica dell'integrità degli oggetti.

Operazioni simultanee di caricamento in più parti

In un ambiente di sviluppo distribuito è possibile che l'applicazione avvii più aggiornamenti sullo stesso oggetto contemporaneamente. L'applicazione potrebbe avviare vari caricamenti in più parti utilizzando la stessa chiave dell'oggetto. Per ciascuno di questi caricamenti, l'applicazione può quindi caricare le parti e inviare una richiesta di completamento del caricamento ad Amazon S3 per creare l'oggetto. Se per i bucket è abilitato il controllo delle versioni S3, il completamento di un caricamento in più parti crea sempre una nuova versione. Per i bucket per i quali non è abilitata la funzione Controllo delle versioni, è possibile che altre richieste ricevute nel periodo di tempo compreso tra l'avvio e il completamento di un caricamento in più parti abbiano la precedenza.

> **Note**
>
> È possibile per altre richieste ricevute nel periodo di tempo compreso tra l'avvio e il completamento di un caricamento in più parti abbiano la precedenza. Ad esempio, se un'altra operazione elimina una chiave dopo l'avvio del caricamento in più parti con tale chiave ma prima del relativo completamento, la risposta relativa al complemento di tale caricamento potrebbe indicare che è stato creato un oggetto senza che sia stato mai visualizzato.

Caricamento in più parti e prezzi

Una volta avviato un caricamento in più parti, Amazon S3 mantiene tutte le parti finché il caricamento non viene completato o interrotto. Per tutta la durata del processo, all'utente vengono fatturati i costi per lo storage, la larghezza di banda e le richieste per questo tipo di caricamento e per le parti associate.

Queste parti vengono addebitate in base alla classe di archiviazione specificata al momento del caricamento delle parti. Restrizioni di parti caricate in S3 Glacier Flexible Retrieval o di S3 Glacier Deep Archive. Le parti in più parti in corso per un'operazione PUT nella classe di archiviazione S3 Glacier Flexible Retrieval vengono fatturate come classe di archiviazione a fasi S3 Glacier Flexible Retrieval a tariffe di archiviazione S3 Standard fino al completamento del caricamento. Inoltre, entrambi UploadPart vengono fatturati alle tariffe CreateMultipartUpload S3 Standard. Solo la CompleteMultipartUpload richiesta viene fatturata alla tariffa S3 Glacier Flexible Retrieval. Analogamente, le parti multiparte in corso per una classe di storage PUT to S3 Glacier Deep Archive vengono fatturate come S3 Glacier Flexible Retrieval Staging Storage alle tariffe di storage S3 Standard fino al completamento del caricamento, con solo la richiesta addebitata alle tariffe di S3 Glacier Deep Archive. CompleteMultipartUpload

Se interrompi il caricamento in più parti, Amazon S3 elimina i manufatti del caricamento e le parti caricate e all'utente non viene più addebitato alcun costo. Non sono previsti costi di cancellazione anticipata per l'eliminazione di caricamenti incompleti in più parti indipendentemente dalla classe di archiviazione specificata. Per ulteriori informazioni sui prezzi, consulta la sezione Prezzi di Amazon S3.

> **Note**
>
> Per ridurre al minimo i costi di archiviazione, ti consigliamo di configurare una regola del ciclo di vita per eliminare i caricamenti in più parti incompleti dopo un numero di giorni specificato

utilizzando l'operazione `AbortIncompleteMultipartUpload`. Per ulteriori informazioni sulla creazione di una regola del ciclo di vita per eliminare i caricamenti in più parti incompleti, consulta Configurazione del ciclo di vita del bucket per l'eliminazione dei caricamenti in più parti incompleti.

Supporto per l'API per il caricamento in più parti

Queste librerie forniscono un'astrazione di alto livello che facilita il caricamento in più parti degli oggetti. Tuttavia, se l'applicazione lo richiede, è possibile utilizzare direttamente REST API. Le sezioni seguenti della Documentazione di riferimento delle API di Amazon Simple Storage Service descrivono REST API per il caricamento in più parti.

Per una procedura dettagliata di caricamento in più parti che utilizza le funzioni AWS Lambda, consulta Caricamento di oggetti di grandi dimensioni su Amazon S3 utilizzando l'accelerazione di caricamento e trasferimento in più parti.

- Creazione di un caricamento in più parti
- Upload Part
- Upload Part (Copy)
- Completamento del caricamento in più parti
- Abort Multipart Upload
- List Parts
- Elenco dei caricamenti in più parti

AWS Command Line Interface supporto per il caricamento in più parti

Gli argomenti seguenti AWS Command Line Interface descrivono le operazioni per il caricamento in più parti.

- Avvio del caricamento in più parti
- Upload Part
- Upload Part (Copy)
- Completamento del caricamento in più parti
- Abort Multipart Upload

- List Parts
- Elenco dei caricamenti in più parti

AWS Supporto SDK per il caricamento in più parti

Puoi utilizzare un AWS SDK per caricare un oggetto in più parti. Per un elenco degli AWS SDK supportati dall'azione dell'API, consulta:

- Creazione di un caricamento in più parti
- Upload Part
- Upload Part (Copy)
- Completamento del caricamento in più parti
- Abort Multipart Upload
- List Parts
- Elenco dei caricamenti in più parti

Autorizzazioni e API per il caricamento in più parti

Per eseguire le operazioni di caricamento in più parti, devi disporre delle autorizzazioni necessarie. Per concedere ai singoli utenti le autorizzazioni per eseguire queste operazioni, è possibile utilizzare le liste di controllo accessi (ACL), la policy di bucket o la policy utente. Nella tabella riportata di seguito sono elencate le autorizzazioni richieste per le varie operazioni di caricamento in più parti quando si utilizzano le liste di controllo accessi (ACL), la policy di bucket o la policy utente.

Operazione	Autorizzazioni richieste
Creazione di un caricamento in più parti	Per creare un caricamento in più parti, è necessario essere autorizzati a eseguire l'operazione `s3:PutObject` su un oggetto. Il proprietario del bucket può consentire ad altre entità di eseguire l'operazione `s3:PutObject` .
Avvio del caricamento in più parti	Per avviare il caricamento in più parti, è necessario essere autorizzati a eseguire l'operazione `s3:PutObject` su un oggetto.

Operazione	Autorizzazioni richieste
	Il proprietario del bucket può consentire ad altre entità di eseguire l'operazione `s3:PutObject` .
Iniziatore	Elemento del container che identifica l'utente che ha avviato il caricamento in più parti. Se l'iniziatore è un Account AWS, questo elemento fornisce le stesse informazioni dell'elemento Owner. Se è un utente IAM, questo elemento fornisce l'ARN e il nome visualizzato dell'utente.
Upload Part	Per caricare una parte, è necessario essere autorizzati a eseguire l'operazione `s3:PutObject` su un oggetto. Il proprietario del bucket deve consentire all'iniziatore di eseguire l'operazione `s3:PutObject` su un oggetto affinché quest'ultimo possa caricare una parte di tale oggetto.
Upload Part (Copy)	Per caricare una parte, è necessario essere autorizzati a eseguire l'operazione `s3:PutObject` su un oggetto. Poiché si sta caricando una parte da un oggetto esistente, è necessario essere autorizzati a eseguire `s3:GetObject` sull'oggetto di origine. Perché l'iniziatore possa caricare una parte di un oggetto, il proprietario del bucket deve consentire all'iniziatore di eseguire l'operazione `s3:PutObject` sull'oggetto.
Completamento del caricamento in più parti	Per completare il caricamento in più parti, è necessario essere autorizzati a eseguire l'operazione `s3:PutObject` su un oggetto. Il proprietario del bucket deve consentire all'iniziatore di eseguire l'operazione `s3:PutObject` su un oggetto affinché quest'ultimo possa completare un caricamento in più parti di tale oggetto.

Operazione	Autorizzazioni richieste
Stop Multipart Upload	Per interrompere un caricamento in più parti, è necessario essere autorizzati a eseguire l'operazione `s3:AbortMultipartUpload` . Di default, il proprietario del bucket e l'iniziatore del caricamento in più parti sono autorizzati a eseguire questa operazione nell'ambito delle policy IAM e del bucket. Se l'iniziatore è un utente IAM, anche a quell'utente Account AWS è consentito interrompere il caricamento in più parti. Con le policy degli endpoint VPC, l'iniziatore del caricamento in più parti non ottiene automaticamente l'autorizzazione a eseguire l'operazione `s3:AbortMultipartUpload` . Oltre a queste impostazioni di default, il proprietario del bucket può consentire ad altre entità di eseguire l'operazione `s3:AbortMultipartUpload` su un oggetto. Il proprietario del bucket può negare a qualsiasi entità di eseguire l'operazione `s3:AbortMultipartUpload` .
List Parts	Per elencare un caricamento in più parti, è necessario essere autorizzati a eseguire l'operazione `s3:ListMultipartUploadParts` . Per default, il proprietario del bucket dispone dell'autorizzazione per elencare le parti per qualsiasi caricamento in più parti nel bucket. L'iniziatore del caricamento in più parti dispone dell'autorizzazione per elencare le parti di un caricamento in più parti specifico. Se l'iniziatore del caricamento in più parti è un utente IAM, l'utente IAM che Account AWS controlla tale utente ha anche l'autorizzazione a elencare parti di tale caricamento. Oltre a queste impostazioni di default, il proprietario del bucket può consentire ad altre entità di eseguire l'operazione `s3:ListMultipartUploadParts` su un oggetto. Il proprietario del bucket può anche negare alle entità l'esecuzione dell'operazione `s3:ListMultipartUploadParts` .

Operazione	Autorizzazioni richieste
Elenco dei caricamenti in più parti	Per elencare i caricamenti in più parti in corso in un bucket, è necessario essere autorizzati a eseguire l'operazione `s3:ListBucketMultipartUploads` su tale bucket. Oltre a queste impostazioni di default, il proprietario del bucket può consentire ad altre entità di eseguire l'operazione `s3:ListBucketMultipartUploads` sul bucket.
AWS KMS Crittografa e decrittografa le autorizzazioni relative alla crittografia e alla decrittografia	Per eseguire un caricamento in più parti con crittografia utilizzando una chiave AWS Key Management Service (AWS KMS) KMS, il richiedente deve disporre dell'autorizzazione e delle azioni sulla chiave. `kms:Decrypt` `kms:GenerateDataKey` Queste autorizzazioni sono obbligatorie perché Amazon S3 deve decrittografare e leggere i dati dalle parti di file crittografate prima di completare il caricamento in più parti. Per ulteriori informazioni, consulta Caricamento di un file di grandi dimensioni su Amazon S3 con la crittografia utilizzando una AWS KMS key nel Knowledge Center di AWS . Se il tuo utente o ruolo IAM coincide con Account AWS la chiave KMS, devi disporre di queste autorizzazioni sulla policy chiave. Se l'utente o il ruolo IAM appartiene a un account diverso rispetto alla chiave KMS, devi disporre delle autorizzazioni sulla policy delle chiavi e sull'utente o sul ruolo IAM.

Per informazioni sulle relazioni tra le autorizzazioni nelle liste di controllo accessi (ACL) e le autorizzazioni nelle policy di accesso, consulta la sezione Mappatura delle autorizzazioni ACL e delle autorizzazioni della policy di accesso. Per informazioni su utenti, ruoli e best practice di IAM, consulta Identità IAM (utenti, gruppi di utenti e ruoli) nella Guida per l'utente di IAM.

Argomenti

- Configurazione del ciclo di vita del bucket per l'eliminazione dei caricamenti in più parti incompleti
- Caricamento di un oggetto utilizzando il caricamento in più parti
- Caricamento di una directory utilizzando la classe.NET di alto livello TransferUtility
- Elenco dei caricamenti in più parti

- Monitoraggio di un caricamento in più parti
- Interruzione di un caricamento in più parti
- Copia di un oggetto utilizzando il caricamento in più parti
- Limiti del caricamenti in più parti di Amazon S3

Configurazione del ciclo di vita del bucket per l'eliminazione dei caricamenti in più parti incompleti

Consigliamo, come best practice, di configurare una regola per il ciclo di vita utilizzando l'operazione `AbortIncompleteMultipartUpload` per ridurre al minimo i costi di archiviazione. Per ulteriori informazioni sull'interruzione di un caricamento in più parti, consulta Interruzione di un caricamento in più parti.

Amazon S3 supporta una regola per il ciclo di vita del bucket che può essere utilizzata per indicare ad Amazon S3 di interrompere i caricamenti in più parti che non sono stati completati entro un determinato numero di giorni dopo l'avvio. Quando un caricamento in più parti non viene completato entro il periodo di tempo specificato, diventa idoneo per un'operazione di interruzione. Quando Amazon S3 interrompe un caricamento in più parti, elimina tutte le parti associate al caricamento in più parti. Questa regola si applica sia ai caricamenti multiparte esistenti sia a quelli creati in un secondo momento.

Di seguito è riportata una configurazione del ciclo di vita di esempio che specifica una regola con l'operazione `AbortIncompleteMultipartUpload`.

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Prefix></Prefix>
        <Status>Enabled</Status>
        <AbortIncompleteMultipartUpload>
          <DaysAfterInitiation>7</DaysAfterInitiation>
        </AbortIncompleteMultipartUpload>
    </Rule>
</LifecycleConfiguration>
```

Nell'esempio, la regola non specifica un valore per l'elemento `Prefix` (prefisso nome della chiave oggetto). Pertanto, la regola viene applicata a tutti gli oggetti nel bucket per i quali sono stati avviati caricamenti in più parti. Tutti i caricamenti in più parti che sono stati avviati e non sono stati completati

entro sette giorni diventano idonei per un'operazione di interruzione. L'azione di interruzione non ha alcun effetto sui caricamenti in più parti completati.

Per ulteriori informazioni sulla configurazione del ciclo di vita dei bucket, consulta Gestione del ciclo di vita dello storage.

> **Note**
>
> Se il caricamento in più parti viene completato entro il numero di giorni specificato nella regola, l'operazione `AbortIncompleteMultipartUpload` del ciclo di vita non viene eseguita e Amazon S3 non intraprende alcuna operazione. Inoltre, questa operazione non si applica agli oggetti. Nessun oggetto viene eliminato da questa operazione del ciclo di vita. Inoltre, non dovrai sostenere costi per l'eliminazione anticipata del ciclo di vita S3 quando rimuovi parti caricate in più parti incomplete.

Utilizzo della console S3

Per gestire automaticamente caricamenti in più parti incompleti, puoi utilizzare la console S3 per creare una regola del ciclo di vita per far scadere byte dei caricamenti in più parti incompleti dal bucket dopo un determinato numero di giorni. Nella seguente procedura viene illustrato come aggiungere una regola del ciclo di vita per eliminare caricamenti in più parti dopo 7 giorni. Per ulteriori informazioni sull'aggiunta di regole del ciclo di vita, consulta Impostazione di una configurazione del ciclo di vita su un bucket.

Per aggiungere una regola del ciclo di vita per interrompere i caricamenti in più parti incompleti che risalgono a più di 7 giorni

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegliere il nome del bucket per il quale si desidera creare una regola del ciclo di vita.
3. Scegliere la scheda Management (Gestione), quindi Create lifecycle rule (Crea regola ciclo di vita).
4. In Lifecycle rule name (Nome regola ciclo di vita) immettere un nome per la regola.

   Il nome deve essere univoco all'interno del bucket.
5. Scegliere l'ambito della regola del ciclo di vita:

- Per creare una regola del ciclo di vita per tutti gli oggetti con un prefisso specifico, scegli Limit the scope of this rule using one or more filters (Limita l'ambito di questa regola utilizzando uno o più filtri) e inserisci il prefisso nel campo Prefix (Prefisso).
- Per applicare una regola del ciclo di vita a tutti gli oggetti nel bucket, scegli This rule applies to all objects in the bucket (Questa regola si applica a tutti gli oggetti nel bucket) e quindi scegli I acknowledge that this rule applies to all objects in the bucket (Confermo che questa regola si applica a tutti gli oggetti nel bucket).

6. In Lifecycle rule actions (Operazioni regola ciclo di vita), seleziona Delete expired object delete markers or incomplete multipart uploads (Elimina contrassegni di eliminazione oggetti scaduti o caricamenti in più parti incompleti).
7. In Delete expired delete markers or incomplete multipart uploads (Elimina contrassegni di eliminazione scaduti o caricamenti in più parti incompleti), seleziona Delete incomplete multipart uploads (Elimina caricamenti in più parti incompleti).
8. Nel campo Number of days (Numero di giorni), inserisci il numero di giorni trascorsi i quali eliminare i caricamenti in più parti incompleti (per questo esempio, 7 giorni).
9. Scegli Crea regola.

Usando il AWS CLI

Il comando seguente `put-bucket-lifecycle-configuration`AWS Command Line Interface (AWS CLI) aggiunge la configurazione del ciclo di vita per il bucket specificato. Per utilizzare questo comando, sostituisci *`user input placeholders`* con le tue informazioni.

```
aws s3api put-bucket-lifecycle-configuration  \
        --bucket example-s3-bucket1  \
        --lifecycle-configuration filename-containing-lifecycle-configuration
```

L'esempio seguente mostra come aggiungere una regola del ciclo di vita per interrompere i caricamenti in più parti incompleti utilizzando la AWS CLI. Include un esempio di configurazione del ciclo di vita JSON per interrompere i caricamenti in più parti incompleti che risalgono a più di 7 giorni.

Per utilizzare i comandi CLI in questo esempio, sostituisci *`user input placeholders`* con le tue informazioni.

Per aggiungere una regola del ciclo di vita per interrompere i caricamenti in più parti incompleti

1. Configura il. AWS CLI Per istruzioni, consulta Sviluppo con Amazon S3 tramite la AWS CLI.

2. Salva la configurazione del ciclo di vita di esempio riportata di seguito in un file (ad esempio, *lifecycle.json*). Questa configurazione di esempio specifica un prefisso vuoto e pertanto non si applica a tutti gli oggetti nel bucket. È possibile specificare un prefisso per limitare la configurazione a un sottoinsieme di oggetti.

```
{
    "Rules": [
        {
            "ID": "Test Rule",
            "Status": "Enabled",
            "Filter": {
                "Prefix": ""
            },
            "AbortIncompleteMultipartUpload": {
                "DaysAfterInitiation": 7
            }
        }
    ]
}
```

3. Esegui il comando della CLI riportato di seguito per impostare la configurazione del ciclo di vita sul bucket.

```
aws s3api put-bucket-lifecycle-configuration   \
--bucket example-s3-bucket1  \
--lifecycle-configuration file://lifecycle.json
```

4. Per verificare che la configurazione del ciclo di vita sia stata impostata sul bucket, recupera la configurazione del ciclo di vita utilizzando il seguente comando `get-bucket-lifecycle`.

```
aws s3api get-bucket-lifecycle  \
--bucket example-s3-bucket1
```

5. Per eliminare la configurazione del ciclo di vita, utilizza il seguente comando `delete-bucket-lifecycle`.

```
aws s3api delete-bucket-lifecycle \
--bucket example-s3-bucket1
```

Caricamento di un oggetto utilizzando il caricamento in più parti

Puoi usare il caricamento in più parti per caricare un singolo oggetto a livello di programmazione su Amazon S3.

Per ulteriori informazioni, consultare le sezioni indicate di seguito.

Utilizzo degli AWS SDK (API di alto livello)

Alcuni AWS SDK prevedono un'API di alto livello che semplifica il caricamento in più parti combinando le diverse operazioni API necessarie per completare un caricamento multiparte in un'unica operazione. Per ulteriori informazioni, consulta Caricamento e copia di oggetti utilizzando il caricamento in più parti.

Se devi mettere in pausa e riprendere i caricamenti in più parti, modificare le dimensioni delle parti durante il caricamento o non conosci in anticipo le dimensioni dei dati, utilizza i metodi API di basso livello. I metodi API di basso livello per i caricamenti in più parti offrono funzionalità aggiuntive, per ulteriori informazioni, consulta. Utilizzo degli AWS SDK (API di basso livello)

Java

Per caricare file di grandi dimensioni, usa la classe. `TransferManager` Questa operazione API di alto livello può caricare dati da un file o da uno stream. nonché impostare opzioni avanzate, come la dimensione delle parti che si intende utilizzare per il caricamento in più parti o il numero di thread contemporanei da utilizzare quando si caricano le parti. È inoltre possibile impostare proprietà dell'oggetto opzionali, la classe di archiviazione o la lista di controllo degli accessi (ACL). Utilizzare le classi `PutObjectRequest` e `TransferManagerConfiguration` per impostare le opzioni avanzate.

Quando possibile, `TransferManager` tenta di utilizzare più thread per caricare più parti di un singolo caricamento alla volta. Quando si ha a che fare con contenuti di grandi dimensioni e larghezza di bada elevata, si può ottenere un aumento significativo in termini di throughput.

Oltre alla funzionalità di caricamento dei file, la classe `TransferManager` permette di interrompere l'esecuzione del caricamento in più parti. Un caricamento è considerato in esecuzione dopo l'avvio, finché non viene completato o interrotto. Il `TransferManager` interrompe tutti i caricamenti in più parti in esecuzione su un bucket specifico avviati prima di una data e un'ora specifiche.

> **Note**
>
> Se l'origine dei dati è costituita da un flusso, la classe `TransferManager` non esegue caricamenti simultanei.

Il seguente esempio di codice mostra come caricare un oggetto utilizzando l'API Java di caricamento in più parti di alto livello (la classe `TransferManager`). Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
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

Per caricare un file in un bucket S3, utilizzare la classe `TransferUtility`. Se si caricano dati da un file, è necessario specificare il nome della chiave dell'oggetto. In caso contrario, l'API utilizza il nome file per il nome della chiave. Durante il caricamento di dati da un flusso, è necessario specificare il nome della chiave dell'oggetto.

Per impostare opzioni di caricamento avanzate, come la dimensione delle parti, il numero di thread durante il caricamento simultaneo di parti, i metadati, la classe di storage o la lista di controllo accessi, utilizza la classe `TransferUtilityUploadRequest`.

> **Note**
>
> Se l'origine dei dati è costituita da un flusso, la classe `TransferUtility` non esegue caricamenti simultanei.

Il seguente esempio di codice C# consente di caricare un file in un bucket Amazon S3 in più parti. Mostra come utilizzare diversi overload `TransferUtility.Upload` per caricare un file. Ciascuna chiamata successiva al caricamento sostituisce il caricamento precedente. Per

informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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

Carica un file di grandi dimensioni.

```
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

Scarica un file di grandi dimensioni.

```
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

Carica un oggetto di grandi dimensioni utilizzando un gestore di caricamento per suddividere i dati in parti e caricarli contemporaneamente.

```
// 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

Scarica un oggetto di grandi dimensioni utilizzando un gestore di download per ottenere i dati in parti e scaricarli contemporaneamente.

```
// 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

Questo argomento spiega come utilizzare la `Aws\S3\Model\MultipartUpload\UploadBuilder` classe di alto livello di AWS SDK for PHP per i caricamenti di file in più parti. Per ulteriori informazioni sull'API AWS SDK for Ruby, AWS vai a SDK for Ruby - Versione 2.

Il seguente esempio del PHP indica come caricare un file in un bucket Amazon S3. L'esempio dimostra come impostare i parametri per l'oggetto `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

Il seguente esempio di codice mostra come caricare un oggetto utilizzando l'API Python di caricamento in più parti di alto livello (la classe `TransferManager`).

```
import sys
import threading

import boto3
from boto3.s3.transfer import TransferConfig


MB = 1024 * 1024
s3 = boto3.resource("s3")


class TransferCallback:
    """
    Handle callbacks from the transfer manager.

    The transfer manager periodically calls the __call__ method throughout
    the upload and download process so that it can take action, such as
    displaying progress to the user and collecting data about the transfer.
    """

    def __init__(self, target_size):
```

```
        self._target_size = target_size
        self._total_transferred = 0
        self._lock = threading.Lock()
        self.thread_info = {}

    def __call__(self, bytes_transferred):
        """
        The callback method that is called by the transfer manager.

        Display progress during file transfer and collect per-thread transfer
        data. This method can be called by multiple threads, so shared instance
        data is protected by a thread lock.
        """
        thread = threading.current_thread()
        with self._lock:
            self._total_transferred += bytes_transferred
            if thread.ident not in self.thread_info.keys():
                self.thread_info[thread.ident] = bytes_transferred
            else:
                self.thread_info[thread.ident] += bytes_transferred

            target = self._target_size * MB
            sys.stdout.write(
                f"\r{self._total_transferred} of {target} transferred "
                f"({(self._total_transferred / target) * 100:.2f}%)."
            )
            sys.stdout.flush()


def upload_with_default_configuration(
    local_file_path, bucket_name, object_key, file_size_mb
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, using the default
    configuration.
    """
    transfer_callback = TransferCallback(file_size_mb)
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def upload_with_chunksize_and_meta(
```

```
    local_file_path, bucket_name, object_key, file_size_mb, metadata=None
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart chunk size and adding metadata to the Amazon S3 object.

    The multipart chunk size controls the size of the chunks of data that are
    sent in the request. A smaller chunk size typically results in the transfer
    manager using more threads for the upload.

    The metadata is a set of key-value pairs that are stored with the object
    in Amazon S3.
    """
    transfer_callback = TransferCallback(file_size_mb)

    config = TransferConfig(multipart_chunksize=1 * MB)
    extra_args = {"Metadata": metadata} if metadata else None
    s3.Bucket(bucket_name).upload_file(
        local_file_path,
        object_key,
        Config=config,
        ExtraArgs=extra_args,
        Callback=transfer_callback,
    )
    return transfer_callback.thread_info


def upload_with_high_threshold(local_file_path, bucket_name, object_key,
 file_size_mb):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart threshold larger than the size of the file.

    Setting a multipart threshold larger than the size of the file results
    in the transfer manager sending the file as a standard upload instead of
    a multipart upload.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info
```

```
def upload_with_sse(
    local_file_path, bucket_name, object_key, file_size_mb, sse_key=None
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, adding server-side
    encryption with customer-provided encryption keys to the object.

    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is
    provided in the download request.
    """
    transfer_callback = TransferCallback(file_size_mb)
    if sse_key:
        extra_args = {"SSECustomerAlgorithm": "AES256", "SSECustomerKey": sse_key}
    else:
        extra_args = None
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, ExtraArgs=extra_args,
 Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_default_configuration(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, using the
    default configuration.
    """
    transfer_callback = TransferCallback(file_size_mb)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_single_thread(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, using a
    single thread.
```

```
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(use_threads=False)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_high_threshold(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, setting a
    multipart threshold larger than the size of the file.

    Setting a multipart threshold larger than the size of the file results
    in the transfer manager sending the file as a standard download instead
    of a multipart download.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_sse(
    bucket_name, object_key, download_file_path, file_size_mb, sse_key
):
    """
    Download a file from an Amazon S3 bucket to a local folder, adding a
    customer-provided encryption key to the request.

    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is
    provided in the download request.
    """
    transfer_callback = TransferCallback(file_size_mb)

    if sse_key:
        extra_args = {"SSECustomerAlgorithm": "AES256", "SSECustomerKey": sse_key}
    else:
```

```
        extra_args = None
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, ExtraArgs=extra_args, Callback=transfer_callback
    )
    return transfer_callback.thread_info
```

Utilizzo degli AWS SDK (API di basso livello)

L' AWS SDK espone un'API di basso livello molto simile all'API REST di Amazon S3 per caricamenti multiparte (vedi. Caricamento e copia di oggetti utilizzando il caricamento in più parti Utilizza l'API di basso livello quando devi mettere in pausa e riprendere i caricamenti in più parti, variare le dimensioni delle parti durante il caricamento o non conosci in anticipo la dimensione dei dati di caricamento. Se non hai questi requisiti, utilizza l'API di alto livello (vedi). Utilizzo degli AWS SDK (API di alto livello)

Java

L'esempio che segue mostra come utilizzare le classi Java di basso livello per il caricamento di un file. Tale esempio esegue i seguenti passaggi:

- Avvia un caricamento in più parti usando il metodo`AmazonS3Client.initiateMultipartUpload()` e passa un oggetto `InitiateMultipartUploadRequest`.
- Salva l'ID di caricamento che viene restituito dal metodo `AmazonS3Client.initiateMultipartUpload()`. Questo ID di caricamento deve essere specificato per ogni operazione di caricamento in più parti successiva.
- Carica le parti dell'oggetto. Per ogni parte, occorre chiamare il metodo `AmazonS3Client.uploadPart()`. Le informazioni sul caricamento della parte devono essere fornite usando un oggetto `UploadPartRequest`.
- Per ogni parte, salva l'ETag dalla risposta del metodo `AmazonS3Client.uploadPart()` in un elenco. I valori dell'ETag vengono utilizzati per completare il caricamento in più parti.
- Chiama il metodo `AmazonS3Client.completeMultipartUpload()` per completare il caricamento in più parti.

Example

Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
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

Il seguente esempio in C# mostra come utilizzare l'API di caricamento AWS SDK for .NET multiparte di basso livello per caricare un file in un bucket S3. Per informazioni sul caricamento in più parti di Amazon S3, consulta Caricamento e copia di oggetti utilizzando il caricamento in più parti.

> **Note**
>
> Quando utilizzi l' AWS SDK for .NET API per caricare oggetti di grandi dimensioni, potrebbe verificarsi un timeout durante la scrittura dei dati nel flusso di richieste. Puoi impostare un timeout esplicito utilizzando la richiesta `UploadPartRequest`.

Il seguente esempio di codice #C mostra come caricare un file in un bucket S3 utilizzando l'API per il caricamento in più parti di basso livello. Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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

Questo argomento mostra come utilizzare il `uploadPart` metodo di basso livello della versione 3 di AWS SDK for PHP per caricare un file in più parti. Per ulteriori informazioni sull'API AWS SDK for Ruby, AWS vai a SDK for Ruby - Versione 2.

Il seguente esempio di codice PHP mostra come caricare un file in un bucket Amazon S3 utilizzando il caricamento in più parti con l'API del PHP di basso livello.

```
 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;
```

Usando il AWS SDK for Ruby

La AWS SDK for Ruby versione 3 supporta i caricamenti multiparte di Amazon S3 in due modi. Il primo metodo prevede la possibilità di utilizzare caricamenti file gestiti. Per ulteriori informazioni, consulta la sezione Caricamento di file in Amazon S3 nel Blog per sviluppatori di AWS . I caricamenti file gestiti rappresentano il metodo consigliato per caricare i file in un bucket. Offrono i seguenti vantaggi:

- Gestiscono i caricamenti in più parti per gli oggetti con una dimensione maggiore di 15 MB.
- Aprono correttamente i file in modalità binaria per evitare problemi di codifica.
- Utilizzano più thread per il caricamento in parallelo delle parti degli oggetti di grandi dimensioni.

In alternativa, è possibile utilizzare direttamente le seguenti operazioni del client di caricamento in più parti:

- create_multipart_upload - Avvia un caricamento in più parti e restituisce un ID di caricamento.
- upload_part - Carica una parte in un caricamento in più parti.
- upload_part_copy - Carica una parte copiando i dati da un oggetto esistente come origine dati.
- complete_multipart_upload - Completa un caricamento in più parti assemblando le parti caricate in precedenza.
- abort_multipart_upload - Interrompe un caricamento in più parti.

Utilizzo di REST API

Le sezioni seguenti della Documentazione di riferimento delle API di Amazon Simple Storage Service descrivono REST API per il caricamento in più parti.

- Avvio del caricamento in più parti
- Upload Part

- Completamento del caricamento in più parti
- Stop Multipart Upload
- List Parts
- Elenco dei caricamenti in più parti

Utilizzando il AWS CLI

Le seguenti sezioni di AWS Command Line Interface (AWS CLI) descrivono le operazioni per il caricamento in più parti.

- Avvio del caricamento in più parti
- Upload Part
- Upload Part (Copy)
- Completamento del caricamento in più parti
- Abort Multipart Upload
- List Parts
- Elenco dei caricamenti in più parti

Puoi inoltre utilizzare REST API per effettuare richieste REST oppure utilizzare uno degli SDK AWS . Per ulteriori informazioni su REST API, consulta Utilizzo di REST API. Per ulteriori informazioni sugli SDK, consulta Caricamento di un oggetto utilizzando il caricamento in più parti.

Caricamento di una directory utilizzando la classe.NET di alto livello TransferUtility

Puoi utilizzare la classe `TransferUtility` per caricare un'intera directory. Per impostazione predefinita, l'API carica solo i file nella posizione root della directory specificata. Tuttavia, puoi specificare il caricamento di file in modo ricorsivo in tutte le sottodirectory.

Per selezionare i file nella directory specificata in base ai criteri di filtro, specificare espressioni di filtro. Ad esempio, per caricare solo i file .pdf da una directory, specificare l'espressione di filtro `"*.pdf"`.

Quando si caricano file da una directory, non è possibile specificare i nomi delle chiavi per l'oggetto risultante. Amazon S3 crea i nomi delle chiavi utilizzando il percorso file originale. Supponiamo, ad esempio, di avere una la directory denominata `c:\myfolder` con la seguente struttura:

Example

```
C:\myfolder
      \a.txt
      \b.pdf
      \media\
             An.mp3
```

Quando effettui un caricamento in questa directory, Amazon S3 utilizza questi nomi della chiave dell'oggetto:

Example

```
a.txt
b.pdf
media/An.mp3
```

Example

Il seguente esempio di codice C# consente di caricare una directory in un bucket Amazon S3. Mostra come utilizzare diversi overload `TransferUtility.UploadDirectory` per caricare la directory. Ciascuna chiamata successiva al caricamento sostituisce il caricamento precedente. Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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);
            }
        }
    }
}
```

Elenco dei caricamenti in più parti

Puoi utilizzare gli AWS SDK (API di basso livello) per recuperare un elenco di caricamenti multiparte in corso in Amazon S3.

Elencare caricamenti in più parti utilizzando l'SDK (API di basso livello) AWS

Java

Le seguenti attività mostrano in dettaglio come utilizzare le classi Java di basso livello per elencare tutti i caricamenti in più parti in corso in un bucket.

Processo di creazione di un elenco di caricamenti in più parti tramite l'API di basso livello

1	Creare un'istanza della classe `ListMultipartUploadsRequest` e specificare il nome del bucket.
2	Esegui il metodo `AmazonS3Client.listMultipartUploads`. Questo metodo restituisce un'istanza della classe `MultipartUploadListing` che fornisce le informazioni sui caricamenti in più parti in corso.

Il seguente esempio di codice Java mostra le attività precedenti.

Example

```
ListMultipartUploadsRequest allMultpartUploadsRequest =
     new ListMultipartUploadsRequest(existingBucketName);
MultipartUploadListing multipartUploadListing =
```

```
    s3Client.listMultipartUploads(allMultpartUploadsRequest);
```

.NET

Per elencare tutti i caricamenti in più parti in corso in uno specifico bucket, utilizza la classe `ListMultipartUploadsRequest` dell'API di basso livello di AWS SDK for .NET per il caricamento in più parti. Il metodo `AmazonS3Client.ListMultipartUploads` restituisce un'istanza della classe `ListMultipartUploadsResponse` che fornisce informazioni sui caricamenti in più parti in corso.

Un caricamento in più parti in corso è un caricamento che è stato avviato utilizzando la richiesta Initiate Multipart Upload, ma che non è ancora stato completato o interrotto. Per ulteriori informazioni sui caricamenti in più parti di Amazon S3, consulta Caricamento e copia di oggetti utilizzando il caricamento in più parti.

Il seguente esempio in C# mostra come utilizzare per AWS SDK for .NET elencare tutti i caricamenti multiparte in corso su un bucket. Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
ListMultipartUploadsRequest request = new ListMultipartUploadsRequest
{
  BucketName = bucketName // Bucket receiving the uploads.
};

ListMultipartUploadsResponse response = await
 AmazonS3Client.ListMultipartUploadsAsync(request);
```

PHP

Questo argomento mostra come utilizzare le classi API di basso livello della versione 3 di AWS SDK for PHP per elencare tutti i caricamenti multiparte in corso su un bucket. Per ulteriori informazioni sull'API AWS SDK for Ruby, AWS vai a SDK for Ruby - Versione 2.

Il seguente esempio di codice PHP mostra come creare un elenco di tutti i caricamenti in più parti in corso in 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());
```

Elenco dei caricamenti in più parti tramite REST API

Le sezioni seguenti della Documentazione di riferimento delle API di Amazon Simple Storage Service descrivono REST API per l'elenco dei caricamenti in più parti.

- ListParts-elenca le parti caricate per un caricamento multiparte specifico.
- ListMultipartUploads-elenca i caricamenti multiparte in corso.

Elencare i caricamenti in più parti utilizzando il AWS CLI

Le seguenti sezioni AWS Command Line Interface descrivono le operazioni per elencare i caricamenti in più parti.

- list-parts: elenca le parti caricate di un caricamento in più parti specifico.
- list-multipart-uploads-Elenca i caricamenti multiparte in corso.

Monitoraggio di un caricamento in più parti

L'API Java per il caricamento in più parti di alto livello fornisce un'interfaccia di ascolto, `ProgressListener`, per il monitoraggio del caricamento di un oggetto in Amazon S3. Gli eventi di stato si verificano periodicamente e inviano al listener la notifica dell'avvenuto trasferimento dei dati.

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

Il seguente codice Java carica un file e utilizza `ProgressListener` per monitorare lo stato del caricamento. Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella Developer Guide. 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

Il seguente esempio di codice C# consente di caricare un file in un bucket S3 utilizzando la classe `TransferUtility` e monitorare lo stato di avanzamento del caricamento. Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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);
        }
    }
}
```

Interruzione di un caricamento in più parti

Dopo aver avviato un caricamento in più parti, le parti vengono caricate. Amazon S3 le archivia, ma crea l'oggetto basato su di esse solo al termine del caricamento di tutte le parti, quindi invia una richiesta `successful` per il completamento del caricamento in più parti (è necessario verificare che questa richiesta vada a buon fine). Quando riceve la richiesta di completamento del caricamento in più parti, Amazon S3 assembla le parti e crea un oggetto. Se il completamento della richiesta di caricamento in più parti non riesce, Amazon S3 non assemblerà le parti e non creerà alcun oggetto.

Ti viene addebitato tutto lo spazio di storage associato alle parti caricate. Per ulteriori informazioni, consulta Caricamento in più parti e prezzi. È pertanto necessario completare il caricamento in più parti per creare l'oggetto oppure interromperlo per rimuovere le parti caricate.

Puoi interrompere un caricamento multiparte in corso in Amazon S3 utilizzando AWS CLI(), AWS Command Line Interface l'API REST o gli SDK. AWS È inoltre possibile interrompere un caricamento in più parti incompleto utilizzando una configurazione del ciclo di vita del bucket.

Utilizzo degli AWS SDK (API di alto livello)

Java

La classe `TransferManager` fornisce il metodo `abortMultipartUploads` per arrestare i caricamenti in più parti in corso. Un caricamento è considerato in esecuzione dopo l'avvio e finché non viene completato o interrotto. Specifica un valore `Date` per fare in modo che l'API interrompa tutti i caricamenti in più parti sul bucket avviati prima del valore specificato per `Date` e ancora in esecuzione.

Le seguenti attività mostrano in dettaglio come utilizzare le classi Java di alto livello per interrompere i caricamenti in più parti.

Processo di interruzione di caricamenti in più parti tramite l'API di alto livello

1	Crea un'istanza della classe `TransferManager` .
2	Esegui il metodo `TransferManager.abortMultipartUploads` passando il nome del bucket e un valore `Date`.

Il codice Java seguente interrompe l'esecuzione di tutti i caricamenti in più parti avviati su un bucket specifico più di una settimana prima. Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
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**
>
> È anche possibile interrompere un caricamento in più parti specifico. Per ulteriori informazioni, consulta Utilizzo degli AWS SDK (API di basso livello).

.NET

L'esempio di codice C# seguente interrompe l'esecuzione di tutti i caricamenti in più parti avviati su un bucket nella settimana precedente. Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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**
>
> È anche possibile interrompere un caricamento in più parti specifico. Per ulteriori informazioni, consulta Utilizzo degli AWS SDK (API di basso livello).

Utilizzo degli AWS SDK (API di basso livello)

È possibile interrompere l'esecuzione di un caricamento in più parti chiamando il metodo `AmazonS3.abortMultipartUpload`. Questo metodo elimina tutte le parti che sono state caricate in Amazon S3 e libera le risorse. È necessario specificare l'ID di caricamento, il nome del bucket e il nome della chiave. Il seguente esempio di codice Java mostra come interrompere l'esecuzione di un caricamento in più parti.

Per interrompere un caricamento in più parti, devi fornire l'ID di caricamento e i nomi di bucket e chiave utilizzati nel caricamento. Dopo aver interrotto un caricamento in più parti, non puoi utilizzare l'ID di caricamento per caricare altre parti. Per ulteriori informazioni sui caricamenti in più parti di Amazon S3, consulta Caricamento e copia di oggetti utilizzando il caricamento in più parti.

Java

Nell'esempio di codice Java seguente viene interrotto un caricamento in più parti in corso.

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**
>
> Invece di interrompere un caricamento in più parti specifico, è possibile interrompere tutti i caricamenti in più parti avviati prima di un orario specifico che sono ancora in corso. Questa operazione di pulizia è utile per interrompere caricamenti in più parti obsoleti che sono stati avviati ma che non sono stati completati o interrotti. Per ulteriori informazioni, consulta Utilizzo degli AWS SDK (API di alto livello).

.NET

L'esempio di codice #C seguente mostra come interrompere un caricamento in più parti. Per un esempio in C# completo che include il codice seguente, consulta Utilizzo degli AWS SDK (API di basso livello).

```
AbortMultipartUploadRequest abortMPURequest = new AbortMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = keyName,
    UploadId = initResponse.UploadId
```

```
};
await AmazonS3Client.AbortMultipartUploadAsync(abortMPURequest);
```

Puoi anche interrompere tutti i caricamenti in più parti in corso che sono stati avviati prima di un determinato orario. Questa operazione di pulizia è utile per interrompere caricamenti in più parti che non sono stati completati o interrotti. Per ulteriori informazioni, consulta Utilizzo degli AWS SDK (API di alto livello).

PHP

Questo esempio mostra come utilizzare una classe della versione 3 di AWS SDK for PHP per interrompere un caricamento multiparte in corso. Per ulteriori informazioni sull'API AWS SDK for Ruby, AWS vai a SDK for Ruby - Versione 2. Nell'esempio il metodo `abortMultipartUpload()`.

Per ulteriori informazioni sull'API AWS SDK for Ruby, AWS vai a SDK for Ruby - Versione 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,
]);
```

Utilizzo di REST API

Per ulteriori informazioni sull'utilizzo dell'API REST per interrompere un caricamento AbortMultipartUploadin più parti, consulta Amazon Simple Storage Service API Reference.

Usando il AWS CLI

Per ulteriori informazioni sull'utilizzo di AWS CLI per interrompere un caricamento in più parti, vedere abort-multipart-uploadnella Guida ai AWS CLI comandi.

Copia di un oggetto utilizzando il caricamento in più parti

Gli esempi in questa sezione mostrano come copiare oggetti con dimensioni superiori a 5 GB utilizzando l'API per il caricamento in più parti. È possibile copiare oggetti con dimensioni inferiori a 5 GB in una sola operazione. Per ulteriori informazioni, consulta Copiare, spostare e rinominare oggetti.

Utilizzo degli SDK AWS

Per copiare un oggetto utilizzando l'API di basso livello, effettua le seguenti operazioni:

- Avvia il caricamento in più parti chiamando il metodo `AmazonS3Client.initiateMultipartUpload()`.
- Salvare l'ID caricamento dall'oggetto della risposta restituito dal metodo `AmazonS3Client.initiateMultipartUpload()`. Si fornisce questo ID di caricamento per ciascuna operazione di caricamento di parte.
- Copia tutte le parti. Per ciascuna parte che è necessario copiare, creare una nuova istanza della classe `CopyPartRequest`. Fornisci le informazioni sulla parte, inclusi i nomi bucket di origine e destinazione, le chiavi dell'oggetto di origine e destinazione, l'ID di caricamento, le posizioni dei primi e degli ultimi byte della parte e il numero della parte.
- Salva le risposte che il metodo `AmazonS3Client.copyPart()` chiama. Ogni risposta include il valore `ETag` e il numero della parte per la parte caricata. Tali informazioni saranno necessarie per completare il caricamento in più parti.
- Chiama il metodo `AmazonS3Client.completeMultipartUpload()` per completare l'operazione di copia.

Java

Example

Nell'esempio Java seguente viene illustrato come utilizzare l'API Java a basso livello Amazon S3 per eseguire una copia in più parti. Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
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

Il seguente esempio in C# mostra come utilizzare AWS SDK for .NET per copiare un oggetto Amazon S3 di dimensioni superiori a 5 GB da una posizione di origine a un'altra, ad esempio da un bucket all'altro. Per copiare gli oggetti con dimensioni inferiori a 5 GB, utilizza la procedura di copia in una sola operazione come descritto in Utilizzo degli AWS SDK. Per ulteriori informazioni sui caricamenti in più parti di Amazon S3, consulta Caricamento e copia di oggetti utilizzando il caricamento in più parti.

Questo esempio mostra come copiare un oggetto Amazon S3 di dimensioni superiori a 5 GB da un bucket S3 a un altro utilizzando l' AWS SDK for .NET API di caricamento multipart.

```
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);
            }
        }
    }
}
```

Utilizzo di REST API

Le sezioni seguenti della Documentazione di riferimento delle API di Amazon Simple Storage Service descrivono REST API per il caricamento in più parti. Per copiare un oggetto esistente, utilizza l'API Upload Part (Copy) e specifica l'oggetto di origine aggiungendo l'intestazione `x-amz-copy-source` nella richiesta.

- Avvio del caricamento in più parti
- Upload Part
- Upload Part (Copy)
- Completamento del caricamento in più parti
- Abort Multipart Upload
- List Parts
- Elenco dei caricamenti in più parti

Si possono utilizzare queste API per effettuare richieste REST personalizzate oppure è possibile utilizzare uno degli SDK forniti da noi. Per ulteriori informazioni sull'utilizzo di Multipart Upload con, consulta. AWS CLIUtilizzando il AWS CLI Per ulteriori informazioni sugli SDK, consulta AWS Supporto SDK per il caricamento in più parti.

Limiti del caricamenti in più parti di Amazon S3

La tabella riportata di seguito fornisce le specifiche di base di un caricamento in più parti. Per ulteriori informazioni, consulta Caricamento e copia di oggetti utilizzando il caricamento in più parti.

Elemento	Specifica
Dimensione massima oggetto	5 TiB
Numero massimo di parti per caricamento	10.000
Numeri delle parti	Da 1 a 10.000 (inclusi)
Dimensione parte	Da 5 MiB a 5 GiB. Non vi è alcun limite minimo di dimensione per l'ultima parte del caricamento in più parti.
Numero massimo di parti restituite per una richiesta di elenco delle parti	1000
Numero massimo di caricamenti in più parti restituiti per una richiesta di elenco dei caricamenti in più parti	1000

Copiare, spostare e rinominare oggetti

L'`CopyObject`operazione crea una copia di un oggetto già archiviato in Amazon S3.

Puoi creare una copia di un oggetto fino a 5 GB in una singola operazione atomica. Tuttavia, per copiare un oggetto di dimensioni superiori a 5 GB, è necessario utilizzare un caricamento in più parti. Per ulteriori informazioni, consulta the section called “Copia di un oggetto”.

L'operazione `CopyObject` consente di effettuare le seguenti operazioni:

- Create copie aggiuntive degli oggetti.
- Rinomina gli oggetti copiandoli ed eliminando quelli originali.
- Copia o sposta gli oggetti da un bucket all'altro, anche trasversalmente Regioni AWS (ad esempio, da a). `us-west-1` `eu-west-2` Quando sposti un oggetto, Amazon S3 copia l'oggetto nella destinazione specificata e quindi elimina l'oggetto di origine.

> **Note**
>
> La copia o lo spostamento di oggetti da una parte all'altra Regioni AWS comporta costi di larghezza di banda. Per ulteriori informazioni, consulta Prezzi di Amazon S3.

- Modifica i metadati degli oggetti. Ogni oggetto Amazon S3 ha metadati. Questi metadati sono un insieme di coppie nome-valore. È possibile impostare i metadati degli oggetti al momento del caricamento di un oggetto. Dopo aver caricato l'oggetto, non è possibile modificare i metadati dell'oggetto. L'unico modo per modificarli è eseguire una copia dell'oggetto e impostare i metadata. A tale scopo, nell'operazione di copia, impostate lo stesso oggetto come origine e destinazione.

  Alcuni metadati degli oggetti sono metadati di sistema, mentre altri sono definiti dall'utente. È possibile controllare alcuni metadati di sistema. Ad esempio, è possibile controllare la classe di archiviazione e il tipo di crittografia lato server da utilizzare per l'oggetto. Quando si copia un oggetto, vengono copiati anche i metadata di sistema controllati dall'utente e i metadata definiti dall'utente. Amazon S3 reimposta i metadati controllati dal sistema. Ad esempio, quando copi un oggetto, Amazon S3 reimposta la data di creazione dell'oggetto copiato. Non è necessario impostare nessuno di questi valori di metadati controllati dal sistema nella richiesta di copia.

  Quando si copia un oggetto, si potrebbe decidere di aggiornare alcuni dei valori dei metadata. Ad esempio, se l'oggetto di origine è configurato per utilizzare l'archiviazione S3 Standard, puoi scegliere di utilizzare S3 Intelligent-Tiering per la copia dell'oggetto. È anche possibile decidere di modificare alcuni dei valori dei metadata definiti dall'utente presenti nell'oggetto di origine. Se scegli di aggiornare uno qualsiasi dei metadati configurabili dall'utente (metadati di sistema o definiti dall'utente) dell'oggetto durante la copia, devi specificare in modo esplicito nella richiesta tutti i metadati configurabili dall'utente presenti nell'oggetto di origine, anche se stai modificando solo uno dei valori dei metadati.

  Per ulteriori informazioni sui metadata degli oggetti, consulta Utilizzo dei metadati degli oggetti.

Copia di oggetti archiviati e ripristinati

Se l'oggetto di origine viene archiviato in Recupero flessibile Amazon S3 Glacier o Deep Archive Amazon S3 Glacier, è necessario ripristinare una copia temporanea prima di poter copiare l'oggetto in un altro bucket. Per ulteriori informazioni sull'archiviazione degli oggetti, consulta la sezione Trasferimento nelle classi di archiviazione S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive (archiviazione di oggetti).

L'operazione Copy nella console Amazon S3 non è supportata per gli oggetti ripristinati nelle classi di storage S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive. Per copiare questi oggetti ripristinati, usa AWS Command Line Interface (AWS CLI), gli AWS SDK o l'API REST di Amazon S3.

Copiare oggetti crittografati

Amazon S3 esegue automaticamente la crittografa di tutti i nuovi oggetti copiati in un bucket S3. Se non si specificano le informazioni di crittografia nella richiesta di copia, la crittografia dell'oggetto di destinazione viene impostata sulla configurazione di crittografia predefinita del bucket di destinazione. Per impostazione predefinita, tutti i bucket hanno un livello di crittografia di base che utilizza la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3). Se il bucket di destinazione ha una configurazione di crittografia predefinita che utilizza la crittografia lato server con una chiave AWS Key Management Service (AWS KMS) (SSE-KMS) o una chiave di crittografia fornita dal cliente (SSE-C), Amazon S3 utilizza la chiave KMS corrispondente o una chiave fornita dal cliente per crittografare la copia dell'oggetto di destinazione.

Quando copi un oggetto, se desideri utilizzare un tipo diverso di impostazione di crittografia per l'oggetto di destinazione, puoi richiedere che Amazon S3 crittografi l'oggetto di destinazione con una chiave KMS, una chiave gestita Amazon S3 o una chiave fornita dal cliente. Se l'impostazione di crittografia nella richiesta è diversa dalla configurazione di crittografia predefinita del bucket di destinazione, l'impostazione di crittografia nella richiesta ha la priorità. Se l'oggetto di origine della copia è crittografato con SSE-C, devi fornire le informazioni di crittografia necessarie nella richiesta in modo che Amazon S3 possa decrittografare l'oggetto per la copia. Per ulteriori informazioni, consulta Protezione dei dati con la crittografia.

Utilizzo dei checksum per la copia di oggetti

Durante la copia di oggetti puoi scegliere di utilizzare un algoritmo di checksum diverso per l'oggetto. Sia che si scelga di utilizzare lo stesso algoritmo o uno nuovo, Amazon S3 calcola un nuovo valore di checksum dopo la copia dell'oggetto. Amazon S3 non copia direttamente il valore del checksum. Il valore del checksum degli oggetti che sono stati caricati utilizzando caricamenti in più parti potrebbe cambiare. Per ulteriori informazioni sul calcolo del checksum, consulta Utilizzo di checksum a livello di parte per caricamenti in più parti.

Copiare più oggetti in un'unica richiesta

Per copiare più di un oggetto Amazon S3 con una singola richiesta, puoi anche utilizzare S3 Batch Operations. Fornisci alle operazioni in batch S3 un elenco di oggetti su cui operare. Le operazioni in batch S3 richiamano la rispettiva API per eseguire l'operazione specificata. Un solo processo di

operazioni in batch può eseguire l'operazione specificata su miliardi di oggetti contenenti esabyte di dati.

La funzionalità S3 Batch Operations tiene traccia dell'avanzamento, invia notifiche e archivia un report di completamento dettagliato di tutte le operazioni, offrendo un'esperienza serverless revisionabile completamente gestita. Puoi utilizzare S3 Batch Operations tramite la console Amazon S3 AWS CLI AWS , gli SDK o l'API REST. Per ulteriori informazioni, consulta the section called “Nozioni di base sulle operazioni in batch”.

Copiare oggetti in bucket di directory

Per informazioni sulla copia di un oggetto in un bucket di directory, vedere. Copia di un oggetto in un bucket di directory Per informazioni sull'utilizzo della classe di storage Amazon S3 Express One Zone con bucket di directory, consulta e. Che cos'è S3 Express One Zone? Bucks di directory

Per copiare un oggetto

Per copiare un oggetto utilizza i metodi riportati di seguito.

Utilizzo della console S3

> **Note**
>
> - Quando si copia un oggetto utilizzando la console Amazon S3, è necessario disporre `s3:ListAllMyBuckets` dell'autorizzazione. La console necessita di questa autorizzazione per convalidare l'operazione di copia. Ad esempio, le politiche che concedono questa autorizzazione, vedithe section called “Esempi di policy basate su identità”.
>
> Se state copiando un oggetto con tag definiti dall'utente, dovete disporre anche dell'`s3:GetObjectTagging`autorizzazione. Se stai copiando un oggetto che non ha tag definiti dall'utente ma ha una dimensione superiore a 16 MB, devi disporre anche dell'autorizzazione s3:. `GetObjectTagging`
>
> Se la policy del bucket di destinazione nega l'`s3:GetObjectTagging`azione, l'oggetto verrà copiato senza i tag definiti dall'utente e riceverai un errore.
> - Gli oggetti crittografati con chiavi di crittografia fornite dal cliente (SSE-C) non possono essere copiati utilizzando la console S3. Per copiare oggetti crittografati con SSE-C, usa l' AWS CLI AWS SDK o l'API REST di Amazon S3.

- La copia tra regioni di oggetti crittografati con SSE-KMS non è supportata dalla console Amazon S3. Per copiare oggetti crittografati con SSE-KMS tra regioni, usa l' AWS SDK o l' AWS CLI API REST di Amazon S3.

Per copiare un oggetto

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Bucket, quindi scegli la scheda Bucket per uso generico. Accedere al bucket o alla cartella Amazon S3 che contiene gli oggetti da copiare.
3. Selezionare la casella di controllo a sinistra dei nomi degli oggetti da copiare.
4. Nel menu Azioni, scegli Copia dall'elenco di opzioni visualizzato.
5. Selezionare il tipo di destinazione e l'account di destinazione. Per specificare il percorso di destinazione, scegliere Browse S3 (Sfoglia S3), passare alla destinazione e selezionare la casella di controllo a sinistra della destinazione. Seleziona Choose destination (Scegli destinazione) nell'angolo in basso a destra.

   In alternativa, immettere il percorso di destinazione.
6. Se la funzione Controllo delle versioni del bucket non è abilitata, potrebbe essere richiesto di confermare se gli oggetti esistenti con lo stesso nome debbano essere sovrascritti. Per confermare questa opzione, selezionare la casella di controllo e continuare. Se si desidera mantenere tutte le versioni degli oggetti in questo bucket, selezionare Enable Bucket Versioning (Abilita funzione Controllo delle versioni del bucket). È anche possibile aggiornare le proprietà di default di crittografia e blocco degli oggetti S3.
7. In Additional checksums (Checksum aggiuntivi) scegli se copiare gli oggetti utilizzando la funzione di checksum esistente o sostituirla con una nuova. Al momento del caricamento degli oggetti hai la possibilità di specificare l'algoritmo di checksum utilizzato per verificare l'integrità dei dati. Quando effettui la copia dell'oggetto hai la possibilità di scegliere una nuova funzione. Se originariamente non hai specificato un checksum aggiuntivo puoi utilizzare questa sezione delle opzioni di copia per aggiungerne uno.

   > **Note**
   >
   > Anche se scegli di utilizzare la stessa funzione di checksum, il valore del checksum potrebbe cambiare se copi un oggetto che ha una dimensione superiore a 16 MB. Il

valore del checksum potrebbe cambiare a causa del modo in cui vengono calcolati i checksum per i caricamenti in più parti. Per ulteriori informazioni su come potrebbe cambiare il checksum durante la copia dell'oggetto, consulta Utilizzo di checksum a livello di parte per caricamenti in più parti.

Per modificare la funzione di checksum, scegli Replace with a new checksum function (Sostituisci con una nuova funzione di checksum). Scegli la nuova funzione di checksum dalla casella. Quando l'oggetto viene copiato, il nuovo checksum viene calcolato e memorizzato utilizzando l'algoritmo specificato.

8. Scegli Copy (Copia) nell'angolo in basso a destra. Amazon S3 copia gli oggetti nella destinazione.

Utilizzo degli AWS SDK

Gli esempi in questa sezione mostrano come copiare gli oggetti con dimensioni superiori a 5 GB in una singola operazione. Per copiare oggetti di dimensioni superiori a 5 GB, devi utilizzare un caricamento in più parti. Per ulteriori informazioni, consulta Copia di un oggetto utilizzando il caricamento in più parti.

Java

Example

Nell'esempio seguente viene illustrato come copiare un oggetto in Amazon S3 tramite la AWS SDK for Java. Per istruzioni su come creare e testare un esempio funzionante, consulta la Guida introduttiva alla AWS SDK for Java Developer Guide.

```
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

Il seguente esempio in C# utilizza l'alto livello AWS SDK for .NET per copiare oggetti di dimensioni fino a 5 GB in un'unica operazione. Per gli oggetti con una dimensione superiore a 5 GB, utilizza l'esempio di copia di un caricamento in più parti descritto in Copia di un oggetto utilizzando il caricamento in più parti.

Questo esempio crea una copia di un oggetto con dimensione massima di 5 GB. Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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

Questo argomento illustra come utilizzare le classi della versione 3 AWS SDK for PHP per copiare un singolo oggetto e più oggetti all'interno di Amazon S3, da un bucket all'altro o all'interno dello stesso bucket.

Per ulteriori informazioni sull'API AWS SDK for Ruby, AWS vai a SDK for Ruby - Versione 2.

Il seguente esempio di PHP illustra l'uso del `copyObject()` metodo per copiare un singolo oggetto all'interno di Amazon S3. Dimostra anche come creare più copie di un oggetto utilizzando un batch di chiamate a `CopyObject` con il metodo. `getcommand()`

Copia di oggetti

1	Crea un'istanza di un client Amazon S3 utilizzando il costruttore della classe `Aws \S3\S3Client` .
2	Per creare più copie di un oggetto, esegui un batch di chiamate al getCommand()metodo client Amazon S3, che viene ereditato dalla classe. Aws\CommandInterface Specificare il comando `CopyObject` come primo argomento e un array contenente il bucket di origine, il nome della chiave di origine, il bucket di destinazione e il nome della chiave di destinazione come secondo argomento.

```
 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

Le seguenti attività ti guidano nell'uso delle Ruby classi per copiare un oggetto in Amazon S3 da un bucket all'altro o all'interno dello stesso bucket.

Copia di oggetti

1	Usa la gemma modularizzata Amazon S3 per la versione 3 di AWS SDK for Ruby, richiedi e fornisci le tue credenziali`aws-sdk-s3` . AWS Per ulteriori informazioni su come fornire le credenziali, consulta Effettuare richieste utilizzando le Account AWS nostre credenziali utente IAM.
2	Fornisci le informazioni della richiesta, come il nome del bucket di origine, il nome del bucket di destinazione, il nome del bucket di destinazione e la chiave di destinazione.

Il seguente esempio di Ruby codice illustra le attività precedenti utilizzando il `#copy_object` metodo per copiare un oggetto da un bucket all'altro.

```
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__
```

Utilizzo di REST API

Questo esempio descrive come copiare un oggetto utilizzando l'API REST di Amazon S3. Per ulteriori informazioni su REST API, consulta CopyObject.

In questo esempio viene copiato l'oggetto `flotsam` dal bucket *`example-s3-bucket1`* all'oggetto `jetsam` del bucket *`example-s3-bucket2`* conservandone i metadata.

```
PUT /jetsam HTTP/1.1
Host: example-s3-bucket2.s3.amazonaws.com
x-amz-copy-source: /example-s3-bucket1/flotsam
```

```
Authorization: AWS AKIAIOSFODNN7EXAMPLE:ENoSbxYByFA0UGLZUqJN5EUnLDg=
Date: Wed, 20 Feb 2008 22:12:21 +0000
```

La firma viene generata dalle seguenti informazioni.

```
PUT\r\n
\r\n
\r\n
Wed, 20 Feb 2008 22:12:21 +0000\r\n

x-amz-copy-source:/example-s3-bucket1/flotsam\r\n
/example-s3-bucket2/jetsam
```

Amazon S3 restituisce la risposta riportata di seguito che specifica l'ETag dell'oggetto e la data dell'ultima modifica.

```
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>
```

Usando il AWS CLI

Puoi anche usare AWS Command Line Interface (AWS CLI) per copiare un oggetto S3. Per ulteriori informazioni, consulta la sezione copy-object nella Documentazione di riferimento della AWS CLI .

Per informazioni su AWS CLI, vedete What is the AWS Command Line Interface? nella Guida AWS Command Line Interface per l'utente.

Spostare un oggetto.

Per spostare un oggetto, utilizzate i seguenti metodi.

Utilizzo della console S3

> **Note**
>
> - Se stai spostando un oggetto con tag definiti dall'utente, devi disporre dell'`s3:GetObjectTagging`autorizzazione. Se stai spostando un oggetto che non ha tag definiti dall'utente ma ha una dimensione superiore a 16 MB, devi avere anche l'autorizzazione s3:. `GetObjectTagging`
>
> Se la policy del bucket di destinazione nega l'`s3:GetObjectTagging`azione, l'oggetto verrà spostato senza i tag definiti dall'utente e riceverai un errore.
> - Gli oggetti crittografati con chiavi di crittografia fornite dal cliente (SSE-C) non possono essere spostati utilizzando la console Amazon S3. Per spostare oggetti crittografati con SSE-C, usa gli AWS SDK o l' AWS CLI API REST di Amazon S3.
> - Quando sposti le cartelle, attendi il completamento dell'operazione di spostamento prima di apportare ulteriori modifiche alle cartelle.
> - Non puoi utilizzare gli alias dei punti di accesso S3 come origine o destinazione per le operazioni Move nella console Amazon S3.

Spostare un oggetto.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Bucket, quindi scegli la scheda Bucket per uso generico. Passare al bucket o alla cartella Amazon S3 che contiene gli oggetti che si desidera spostare.
3. Selezionare la casella di controllo a sinistra dei nomi degli oggetti che si desidera spostare.
4. Nel menu Azioni, scegli Sposta.
5. Per specificare il percorso di destinazione, scegliere Browse S3 (Sfoglia S3), passare alla destinazione e selezionare la casella di controllo a sinistra della destinazione. Seleziona Choose destination (Scegli destinazione) nell'angolo in basso a destra.

In alternativa, immettere il percorso di destinazione.

6. Se la funzione Controllo delle versioni del bucket non è abilitata, potrebbe essere richiesto di confermare se gli oggetti esistenti con lo stesso nome debbano essere sovrascritti. Per confermare questa opzione, selezionare la casella di controllo e continuare. Se si desidera mantenere tutte le versioni degli oggetti in questo bucket, selezionare Enable Bucket Versioning (Abilita funzione Controllo delle versioni del bucket). È anche possibile aggiornare le proprietà di default di crittografia e blocco degli oggetti.
7. Scegli Copy (Copia) nell'angolo in basso a destra. Amazon S3 sposta i tuoi oggetti nella destinazione.

> **Note**
>
> - Questa operazione crea una copia di tutti gli oggetti specificati con impostazioni aggiornate, aggiorna la data dell'ultima modifica nella posizione specificata e aggiunge un contrassegno di eliminazione all'oggetto originale.
> - Questa operazione aggiorna i metadati per la funzione Controllo versioni del bucket, la crittografia, le caratteristiche di blocco degli oggetti e gli oggetti archiviati.

Usando il AWS CLI

Puoi anche usare il AWS Command Line Interface (AWS CLI) per spostare un oggetto S3. Per ulteriori informazioni, consulta la sezione mv nella Documentazione di riferimento della AWS CLI .

Per informazioni su AWS CLI, vedete What is the AWS Command Line Interface? nella Guida AWS Command Line Interface per l'utente.

Per rinominare un oggetto

Per rinominare un oggetto, utilizzare la procedura seguente.

> **Note**
>
> - La ridenominazione di un oggetto crea una copia dell'oggetto con una nuova data dell'ultima modifica, quindi aggiunge un indicatore di eliminazione all'oggetto originale.

- Le impostazioni del bucket per la crittografia predefinita vengono applicate automaticamente a qualsiasi oggetto specificato non crittografato.
- Non puoi utilizzare la console Amazon S3 per rinominare oggetti con chiavi di crittografia fornite dal cliente (SSE-C). Per rinominare oggetti crittografati con SSE-C, usa gli AWS CLI AWS SDK o l'API REST di Amazon S3 per copiare tali oggetti con nuovi nomi.
- Se questo bucket utilizza l'impostazione imposta dal proprietario del bucket per S3 Object Ownership, le liste di controllo degli accessi agli oggetti (ACL) non verranno copiate.
- Se stai rinominando un oggetto con tag definiti dall'utente, devi disporre dell'autorizzazione. `s3:GetObjectTagging` Se stai rinominando un oggetto che non ha tag definiti dall'utente ma ha una dimensione superiore a 16 MB, devi avere anche l'autorizzazione s3:. `GetObjectTagging`

  Se la policy del bucket di destinazione nega l'`s3:GetObjectTaggingazione`, l'oggetto verrà rinominato, ma i tag definiti dall'utente verranno rimossi dall'oggetto e verrà visualizzato un errore.

Per rinominare un oggetto

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Bucket, quindi scegli la scheda Bucket per uso generico. Accedi al bucket o alla cartella Amazon S3 che contiene l'oggetto che desideri rinominare.
3. Seleziona la casella di controllo a sinistra del nome dell'oggetto che desideri rinominare.
4. Nel menu Azioni, scegli Rinomina oggetto.
5. Nella casella Nuovo nome oggetto, inserite il nuovo nome per l'oggetto.
6. Scegli Salva modifiche nell'angolo in basso a destra. Amazon S3 rinomina il tuo oggetto.

Download di oggetti

In questa sezione viene illustrato come scaricare oggetti da un bucket Amazon S3. Con Amazon S3, puoi archiviare oggetti in uno o più bucket e ogni singolo oggetto può avere dimensioni fino a 5 TB. Qualsiasi oggetto Amazon S3 non archiviato è accessibile in tempo reale. Gli oggetti archiviati,

tuttavia, devono essere ripristinati prima di poter essere scaricati. Per ulteriori informazioni sul download di oggetti archiviati, consulta the section called “Download di oggetti archiviati”.

Puoi scaricare un singolo oggetto utilizzando la console Amazon S3, AWS Command Line Interface (AWS CLI), gli AWS SDK o l'API REST di Amazon S3. Per scaricare un oggetto da S3 senza scrivere alcun codice o eseguire comandi, usa la console S3. Per ulteriori informazioni, consulta the section called “Download di un oggetto”.

Per scaricare più oggetti, usa AWS CloudShell AWS CLI, o gli SDK. AWS Per ulteriori informazioni, consulta the section called “Download di più oggetti”.

Se devi scaricare parte di un oggetto, utilizza parametri aggiuntivi con l'API AWS CLI o REST per specificare solo i byte che desideri scaricare. Per ulteriori informazioni, consulta the section called “Download di parte di un oggetto”.

Se devi scaricare un oggetto di cui non sei il proprietario, chiedi al proprietario dell'oggetto di generare un URL prefirmato che ti consenta di scaricare l'oggetto. Per ulteriori informazioni, consulta the section called “Download di un oggetto da un altro Account AWS”.

Quando scarichi oggetti al di fuori della AWS rete, vengono applicate le tariffe per il trasferimento dei dati. Il trasferimento di dati all'interno della AWS rete è gratuito all'interno della stessa Regione AWS, ma eventuali `GET` richieste verranno addebitate all'utente. Per ulteriori informazioni sui costi del trasferimento dei dati e le tariffe di recupero dei dati, consulta Prezzi di Amazon S3.

Argomenti

Download di un oggetto

Puoi scaricare un oggetto utilizzando la console Amazon S3 AWS CLI, gli AWS SDK o l'API REST.

Utilizzo della console S3

In questa sezione viene illustrato come utilizzare la console Amazon S3 per scaricare un oggetto da un bucket S3.

> **Note**
>
> - Puoi scaricare un solo oggetto alla volta.
> - Se utilizzi la console di Amazon S3 per scaricare un oggetto il cui nome della chiave termina con un punto (`.`), il punto viene rimosso dal nome della chiave dell'oggetto scaricato. Per conservare il punto alla fine del nome dell'oggetto scaricato, devi utilizzare AWS Command Line Interface (AWS CLI), gli AWS SDK o l'API REST di Amazon S3.

Per scaricare un oggetto da un bucket S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegliere il nome del bucket dal quale si desidera scaricare un oggetto.
3. È possibile scaricare un oggetto da un bucket S3 in uno qualsiasi dei modi seguenti:
   - Seleziona la casella di controllo accanto all'oggetto e scegli Scarica. Se desideri scaricare l'oggetto in una cartella specifica, nel menu Azioni, scegli Scarica come.
   - Se desideri scaricare una versione specifica dell'oggetto, attiva Mostra versioni (che si trova accanto alla casella di ricerca). Seleziona la casella di controllo accanto alla versione dell'oggetto desiderato e scegli Scarica. Se desideri scaricare l'oggetto in una cartella specifica, nel menu Azioni, scegli Scarica come.

Usando il AWS CLI

L'esempio `get-object` seguente mostra come utilizzare la AWS CLI per scaricare un oggetto da Amazon S3. Questo comando recupera l'oggetto *`folder/my_image`* dal bucket *`example-s3-bucket1`*. L'oggetto verrà scaricato in un file denominato *`my_downloaded_image`*.

```
aws s3api get-object --bucket example-s3-bucket1 --key folder/
my_image my_downloaded_image
```

Per ulteriori informazioni ed esempi, consulta get-object nel Riferimento ai comandi AWS CLI .

Utilizzo degli AWS SDK

Per esempi su come scaricare un oggetto con gli AWS SDK, consulta. Utilizzo `GetObject` con un AWS SDK o una CLI

Per informazioni generali sull'utilizzo di diversi AWS SDK, consulta. Sviluppo con Amazon S3 utilizzando gli SDK AWS

Utilizzo di REST API

Puoi utilizzare REST API per recuperare oggetti da Amazon S3. Per ulteriori informazioni, consulta GetObject in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Download di più oggetti

Puoi scaricare più oggetti utilizzando AWS CloudShell AWS CLI, o gli AWS SDK.

Usando AWS CloudShell in AWS Management Console

AWS CloudShell è una shell preautenticata basata su browser che è possibile avviare direttamente da. AWS Management Console

Per ulteriori informazioni su AWS CloudShell, consulta What is? CloudShell nella Guida AWS CloudShell per l'utente.

> **Important**
>
> Con AWS CloudShell, la tua home directory ha uno spazio di archiviazione fino a 1 GB per. Regione AWS Pertanto non puoi sincronizzare i bucket con oggetti per un totale superiore a tale quantità. Per ulteriori limitazioni, consulta Service quotas e restrizioni nella Guida per l'utente di AWS CloudShell .

Per scaricare oggetti utilizzando AWS CloudShell

1. Accedere AWS Management Console e aprire la CloudShell console all'indirizzo https://console.aws.amazon.com/cloudshell/.
2. Esegui il comando seguente per sincronizzare gli oggetti nel tuo bucket con CloudShell. Il comando seguente sincronizza gli oggetti dal bucket denominato *example-s3-bucket1* e crea una cartella denominata in. *temp* CloudShell CloudShell sincronizza gli oggetti con questa cartella. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue specifiche informazioni.

   ```
   aws s3 sync s3://example-s3-bucket1 ./temp
   ```

   > **Note**
   >
   > Per eseguire la corrispondenza del modello per escludere o includere oggetti particolari, puoi utilizzare i parametri `--exclude "value"` e `--include "value"` con il comando `sync`.

3. Esegui il comando seguente per comprimere gli oggetti nella cartella denominata *temp* in un file denominato *temp.zip*.

   ```
   zip temp.zip -r temp/
   ```

4. Scegli Azioni, quindi seleziona Scarica file.
5. Immetti un nome file **`temp.zip`**, quindi scegli Scarica.
6. (Facoltativo) Eliminare il *temp.zip* file e gli oggetti sincronizzati con la *temp* cartella in. CloudShell Con AWS CloudShell, disponi di un'archiviazione persistente fino a 1 GB per ciascuna Regione AWS.

   Puoi utilizzare il seguente comando di esempio per eliminare il file `.zip` e la cartella. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

   ```
   rm temp.zip && rm -rf temp/
   ```

Usando il AWS CLI

L'esempio seguente mostra come utilizzare il AWS CLI per scaricare tutti i file o gli oggetti contenuti nella directory o nel prefisso specificati. Questo comando copia tutti gli oggetti dal bucket *example-s3-bucket1* nella directory corrente. Per utilizzare questo comando di esempio, usa il nome del bucket al posto di *example-s3-bucket1*.

```
aws s3 cp s3://example-s3-bucket1 . --recursive
```

Il comando seguente scarica tutti gli oggetti sotto il prefisso *logs* nel bucket *example-s3-bucket1* nella directory corrente. Inoltre, utilizza i parametri `--exclude` e `--include` per copiare solo gli oggetti con il suffisso *.log*. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
aws s3 cp s3://example-s3-bucket1/logs/ . --recursive --exclude "*" --include "*.log"
```

Per ulteriori informazioni ed esempi, consulta cp nel Riferimento ai comandi AWS CLI .

Utilizzo degli SDK AWS

Per esempi su come scaricare tutti gli oggetti in un bucket Amazon S3 con gli AWS SDK, consulta. Scaricare tutti gli oggetti da un bucket Amazon Simple Storage Service (Amazon S3) in una directory locale

Per informazioni generali sull'utilizzo di diversi AWS SDK, consulta. Sviluppo con Amazon S3 utilizzando gli SDK AWS

Download di parte di un oggetto

Puoi scaricare parte di un oggetto utilizzando la AWS CLI nostra API REST. A tale scopo, utilizza parametri aggiuntivi per specificare la parte di un oggetto da scaricare.

Utilizzando il AWS CLI

Il comando di esempio seguente esegue una richiesta `GET` per un intervallo di byte nell'oggetto denominato *folder/my_data* nel bucket denominato *example-s3-bucket1*. Nella richiesta, l'intervallo di byte deve essere preceduto da `bytes=`. L'oggetto parziale viene scaricato nel file di output denominato *my_data_range*. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
aws s3api get-object --bucket example-s3-bucket1 --key folder/my_data --range
 bytes=0-500 my_data_range
```

Per ulteriori informazioni ed esempi, consulta get-object nel Riferimento ai comandi AWS CLI .

Per ulteriori informazioni sull'intestazione `Range` HTTP, consulta RFC 9110 nel sito Web RFC Editor.

> **Note**
>
> Amazon S3 non supporta il recupero di più intervalli di dati in una singola richiesta `GET`.

Utilizzo di REST API

Puoi utilizzare i parametri `partNumber` e `Range` nella REST API per recuperare parti di oggetti da Amazon S3. Per ulteriori informazioni, consulta GetObject in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Download di un oggetto da un altro Account AWS

Per concedere un accesso limitato nel tempo agli oggetti senza aggiornare la policy del bucket, puoi utilizzare un URL prefirmato.

Un URL prefirmato può essere inserito in un browser o utilizzato da un programma per scaricare un oggetto. Le credenziali utilizzate dall'URL sono quelle dell' AWS utente che ha generato l'URL. Dopo che l'URL viene creato, chiunque disponga dell'URL prefirmato può scaricare l'oggetto corrispondente fino alla scadenza dell'URL.

Utilizzo di un URL prefirmato nella console S3

Puoi utilizzare la console Amazon S3 per generare un URL prefirmato per un oggetto seguendo questi fasi. Quando si utilizza la console, il tempo massimo di scadenza per un URL prefirmato è di 12 ore dal momento della creazione.

Generazione di un URL prefirmato utilizzando la console di Amazon S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).

3. Nell'elenco Buckets (Bucket) scegli il nome del bucket contenente gli oggetto per cui desideri ottenere l'URL prefirmato.
4. Nell'elenco Objects (Oggetti), seleziona l'oggetto per cui desideri creare un URL prefirmato.
5. Nel menu Operazioni oggetti, scegli Crea URL prefirmato.
6. Specifica per quanto tempo desideri che l'URL prefirmato sia valido.
7. Scegli Create presigned URL (Crea URL prefirmato).
8. Quando viene visualizzato un messaggio di conferma, l'URL viene automaticamente copiato negli appunti. Verrà visualizzato un pulsante per copiare l'URL preimpostato qualora fosse necessario copiarlo di nuovo.
9. Per scaricare l'oggetto, incolla l'URL in qualsiasi browser; l'oggetto tenterà di scaricarlo.

Per ulteriori informazioni sugli URL prefirmati e altri metodi per crearli, consulta Utilizzo di URL prefirmati.

Download di oggetti archiviati

Per ridurre i costi di archiviazione degli oggetti a cui si accede raramente, è possibile archiviare tali oggetti. Quando si archivia un oggetto, questo viene spostato in una archiviazione a basso costo, il che significa che non è possibile accedervi in tempo reale. Per scaricare un oggetto archiviato, occorre prima ripristinarlo.

Puoi ripristinare oggetti archiviati in pochi minuti o ore, a seconda della classe di storage. Puoi ripristinare un oggetto archiviato utilizzando la console Amazon S3, S3 Batch Operations, l'API REST di Amazon S3, AWS gli SDK e (). AWS Command Line Interface AWS CLI

Per istruzioni, consulta Ripristino di un oggetto archiviato. Dopo aver ripristinato l'oggetto archiviato, puoi scaricarlo.

Risoluzione dei problemi di download degli oggetti

Autorizzazioni insufficienti o policy utente errate per bucket o AWS Identity and Access Management (IAM) possono causare errori quando si tenta di scaricare oggetti da Amazon S3. Questi problemi possono spesso causare errori di accesso negato (403 Forbidden), in cui Amazon S3 non è in grado di consentire l'accesso a una risorsa.

Per cause comuni di errori di accesso negato (403 Forbidden), consulta Risoluzione dei problemi relativi agli errori di accesso negato (403 Accesso negato) in Amazon S3.

Verifica dell'integrità degli oggetti

Amazon S3 utilizza i valori di checksum per verificare l'integrità dei dati da caricare o scaricare da Amazon S3. Inoltre, puoi richiedere che venga calcolato un altro valore di checksum per qualsiasi oggetto da archiviare in Amazon S3. Puoi selezionare uno dei diversi algoritmi di checksum per utilizzarlo durante il caricamento o la copia dei dati. Amazon S3 usa questo algoritmo per calcolare un valore di checksum aggiuntivo e archiviarlo come parte dei metadati dell'oggetto. Per ulteriori informazioni su come utilizzare checksum aggiuntivi per verificare l'integrità dei dati, consulta Tutorial: Verifica dell'integrità dei dati in Amazon S3 con checksum aggiuntivi.

Quando carichi un oggetto, puoi facoltativamente includere un checksum precalcolato come parte della richiesta. Amazon S3 confronta il checksum fornito con il checksum calcolato utilizzando l'algoritmo specificato. Se i due valori non corrispondono, Amazon S3 genera un errore.

Utilizzo di algoritmi di checksum supportati

Amazon S3 ti offre la possibilità di scegliere l'algoritmo di checksum da usare per convalidare i tuoi dati durante il caricamento o il download. Puoi selezionare uno dei seguenti algoritmi di controllo dell'integrità dei dati Secure Hash Algorithms (SHA) o Cyclic Redundancy Check (CRC):

- CRC32
- CRC32C
- SHA-1
- SHA-256

Quando carichi un oggetto, puoi specificare l'algoritmo da utilizzare:

- Quando utilizzi il AWS Management Console, selezioni l'algoritmo di checksum che desideri utilizzare. Puoi inoltre facoltativamente specificare il valore di checksum dell'oggetto. Quando Amazon S3 riceve l'oggetto, calcola il checksum utilizzando l'algoritmo specificato. Se i due valori di checksum non corrispondono, Amazon S3 genera un errore.
- Quando usi un SDK puoi impostare il valore del parametro `x-amz-sdk-checksum-algorithm`sull'algoritmo che Amazon S3 deve utilizzare per il calcolo del checksum. Amazon S3 calcola automaticamente il valore del checksum.
- Quando usi REST API non utilizzi il parametro `x-amz-sdk-checksum-algorithm`. Usi invece una delle intestazioni specifiche dell'algoritmo (ad esempio, `x-amz-checksum-crc32`).

Per ulteriori informazioni sul caricamento degli oggetti, consulta Caricamento degli oggetti.

Per applicare uno di questi valori di checksum agli oggetti già caricati su Amazon S3, puoi copiare l'oggetto. Quando copi un oggetto puoi specificare se vuoi utilizzare l'algoritmo di checksum esistente o utilizzarne uno nuovo. Puoi specificare un algoritmo di checksum quando si utilizza qualsiasi meccanismo supportato per la copia di oggetti, incluso S3 Batch Operations (Operazioni di batch S3). Per ulteriori informazioni sulle operazioni in batch, consulta Esecuzione di operazioni in batch su larga scala su oggetti Amazon S3.

> **Important**
>
> Se utilizzi un caricamento in più parti con checksum aggiuntivi, i numeri delle parti in più parti devono essere consecutivi. Quando usi i checksum aggiuntivi, se tenti di completare una richiesta di caricamento in più parti con numeri parte non consecutivi, Amazon S3 genera un errore HTTP `500 Internal Server Error`.

Dopo aver caricato gli oggetti puoi ottenere il valore di checksum e confrontarlo con un valore di checksum precalcolato o precedentemente archiviato calcolato utilizzando lo stesso algoritmo.

Utilizzo della console S3

Per ulteriori informazioni sull'utilizzo della console e sulla specifica degli algoritmi di checksum da utilizzare durante il caricamento degli oggetti, consultare Caricamento degli oggetti e Tutorial: Verifica dell'integrità dei dati in Amazon S3 con checksum aggiuntivi.

Utilizzo degli SDK AWS

L'esempio seguente mostra come utilizzare gli AWS SDK per caricare un file di grandi dimensioni con caricamento in più parti, scaricare un file di grandi dimensioni e convalidare un file di caricamento composto da più parti, il tutto utilizzando SHA-256 per la convalida dei file.

Java

Example Esempio: caricamento, download e verifica di un file di grandi dimensioni con SHA-256

Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella Developer Guide. 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();
            }
        }
```

```
    }
```

Utilizzo di REST API

Puoi inviare richieste REST per caricare un oggetto con un valore di checksum con PutObjectcui verificare l'integrità dei dati. Puoi anche recuperare il valore di checksum per gli oggetti utilizzando o. GetObjectHeadObject

Usando il AWS CLI

È possibile inviare una richiesta PUT per caricare un oggetto di un massimo di 5 GB in una singola operazione. Per ulteriori informazioni, consulta `PutObject` nella Documentazione di riferimento per i comandi di AWS CLI . Puoi utilizzare anche `get-object` e `head-object` per recuperare il checksum di un oggetto già caricato per verificare l'integrità dei dati.

Per informazioni, consulta le domande frequenti sulla CLI di Amazon S3 nella Guida per l'utente.AWS Command Line Interface

Utilizzo di Content-MD5 durante il caricamento di oggetti

Un altro modo per verificare l'integrità dell'oggetto dopo il caricamento consiste nel fornire un digest MD5 dell'oggetto durante il caricamento. Se calcoli il digest MD5 dell'oggetto, puoi fornire il digest con il comando PUTutilizzando l'intestazione `Content-MD5`.

Dopo aver caricato l'oggetto, Amazon S3 calcola il digest MD5 dell'oggetto e lo confronta con il valore fornito. La richiesta ha esito positivo solo se i due digest corrispondono.

Non è obbligatorio fornire un digest MD5, ma è possibile utilizzarlo per verificare l'integrità dell'oggetto come parte del processo di caricamento.

Utilizzo di Content-MD5 e di ETag per verificare gli oggetti caricati

Il tag di entità (ETag) di un oggetto rappresenta una versione specifica di tale oggetto. L'ETag riflette solo i cambiamenti al contenuto di un oggetto, non i suoi metadati. Se cambiano solo i metadati di un oggetto, l'ETag rimane invariato.

In base all'oggetto, l'ETag dell'oggetto potrebbe essere un digest MD5 dei dati dell'oggetto:

- Se un oggetto viene creato dall'operazione `PutObject`, `PostObject` o `CopyObject` oppure attraverso la AWS Management Console e l'oggetto è anche in testo normale o crittografato tramite

la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3), l'oggetto ha un ETag che è un digest MD5 dei dati dell'oggetto.

- Se un oggetto viene creato dall'`CopyObject`operazione`PutObject`, o o tramite la `PostObject` AWS Management Console, e tale oggetto è crittografato mediante crittografia lato server con chiavi fornite dal cliente (SSE-C) o crittografia lato server con chiavi () (SSE-KMS), quell'oggetto ha un ETag che non è un digest MD5 dei suoi dati oggetto. AWS Key Management Service AWS KMS
- Se un oggetto viene creato dall'operazione `Multipart Upload` o `Part Copy`, l'ETag dell'oggetto non è un digest MD5, indipendentemente dal metodo di crittografia. Se un oggetto è più grande di 16 MB, la AWS Management Console carica o copia l'oggetto come caricamento in più parti e quindi l'ETag non è un digest MD5.

Per gli oggetti in cui l'ETag è il digest `Content-MD5` dell'oggetto, puoi confrontare il valore ETag dell'oggetto con un digest `Content-MD5` calcolato o precedentemente archiviato.

Utilizzo dei checksum finali

Quando carichi oggetti su Amazon S3, puoi fornire un checksum precalcolato per l'oggetto o utilizzare AWS un SDK per creare automaticamente checksum finali per tuo conto. Se decidi di utilizzare un checksum finale, Amazon S3 genera automaticamente il checksum utilizzando l'algoritmo specificato e lo utilizza per convalidare l'integrità dell'oggetto durante il caricamento.

Per creare un checksum finale quando usi un SDK, compila il parametro con il tuo algoritmo preferito AWS . `ChecksumAlgorithm` L'SDK utilizza l'algoritmo per calcolare il checksum dell'oggetto (o delle parti dell'oggetto) e lo aggiunge automaticamente alla fine della richiesta di caricamento. Questo comportamento ti consente di risparmiare tempo perché Amazon S3 esegue sia la verifica che il caricamento dei tuoi dati in un unico passaggio.

> **Important**
>
> Se utilizzi S3 Object Lambda, tutte le richieste a S3 Object Lambda vengono firmate tramite `s3-object-lambda` anziché `s3`. Questo comportamento influisce sulla firma dei valori di checksum finali. Per ulteriori informazioni su S3 Object Lambda, consulta Trasformazione di oggetti con S3 Object Lambda.

Utilizzo di checksum a livello di parte per caricamenti in più parti

È possibile caricare gli oggetti in Amazon S3 come un singolo oggetto o tramite il processo di caricamento in più parti. Gli oggetti di dimensioni superiori a 16 MB che vengono caricati tramite la console, vengono caricati automaticamente utilizzando i caricamenti in più parti. Per ulteriori informazioni sui caricamenti in più parti, consulta la sezione Caricamento e copia di oggetti utilizzando il caricamento in più parti.

Quando un oggetto viene caricato come caricamento in più parti, l'ETag dell'oggetto non è un digest MD5 dell'intero oggetto. Amazon S3 calcola il digest MD5 di ogni singola parte durante il caricamento. I digest MD5 vengono utilizzati per determinare l'ETag dell'oggetto finale. Amazon S3 concatena i byte per i digest MD5 e quindi calcola il digest MD5 di questi valori concatenati. Il passaggio finale della creazione dell'ETag consiste nell'aggiunta da parte di Amazon S3 di un trattino con il numero totale di parti alla fine.

Ad esempio, considera un oggetto caricato con un caricamento in più parti con un ETag di `C9A5A6878D97B48CC965C1E41859F034-14`. In questo caso, `C9A5A6878D97B48CC965C1E41859F034` è il digest MD5 di tutti i digest concatenati. La `-14` indica che ci sono 14 parti associate al caricamento in più parti di questo oggetto.

Se hai abilitato ulteriori valori di checksum per il tuo oggetto in più parti, Amazon S3 calcola il checksum di ogni singola parte utilizzando l'algoritmo di checksum specificato. Il checksum per l'oggetto completato viene calcolato nello stesso modo in cui Amazon S3 calcola il digest MD5 per il caricamento in più parti. È possibile utilizzare questo checksum per verificare l'integrità dell'oggetto.

Per recuperare informazioni sull'oggetto, incluso il numero di parti che compongono l'intero oggetto, puoi utilizzare l'operazione. GetObjectAttributes Con i checksum aggiuntivi, puoi recuperare anche le informazioni di ogni singola parte, incluso il valore di checksum di ciascuna parte.

Per i caricamenti completati, potete ottenere il checksum di una singola parte utilizzando le HeadObjectoperazioni GetObjecto e specificando un numero di parte o un intervallo di byte allineato a una singola parte. Se desideri recuperare i valori del checksum per le singole parti dei caricamenti in più parti ancora in corso, puoi utilizzare. ListParts

A causa del modo in cui Amazon S3 calcola il checksum per gli oggetti in più parti, il valore del checksum dell'oggetto potrebbe cambiare se lo si copia. Se utilizzi un SDK o l'API REST e effettui una chiamata CopyObject, Amazon S3 copia qualsiasi oggetto fino ai limiti di dimensione `CopyObject` dell'operazione API. Amazon S3 esegue questa copia come un'unica operazione, indipendentemente dal fatto che l'oggetto sia stato caricato in una singola richiesta o come parte di

un caricamento in più parti. Con il comando copy, il checksum dell'oggetto è un checksum diretto dell'oggetto completo. Se l'oggetto è stato originariamente caricato utilizzando un caricamento in più parti, il valore del checksum cambia anche se i dati non sono stati modificati.

> **Note**
>
> Gli oggetti che sono più grandi delle limitazioni di dimensioni dell'operazione API `CopyObject` devono utilizzare i comandi di copia in più parti.

> **Important**
>
> Quando esegui alcune operazioni utilizzando AWS Management Console, Amazon S3 utilizza un caricamento in più parti se l'oggetto ha una dimensione superiore a 16 MB. In questo caso, il checksum non è un checksum diretto dell'oggetto completo, ma piuttosto un calcolo basato sui valori di checksum di ogni singola parte.
> Ad esempio, considera un oggetto di dimensioni di 100 MB che hai caricato come caricamento diretto di singola parte utilizzando REST API. Il checksum in questo caso è un checksum dell'intero oggetto. Se successivamente utilizzi la console per rinominare l'oggetto, copiarlo, modificare la classe di storage o i metadati, Amazon S3 utilizza la funzionalità di caricamento in più parti per aggiornare l'oggetto. Di conseguenza, Amazon S3 crea un nuovo valore di checksum per l'oggetto calcolato in base ai valori di checksum delle singole parti. L'elenco precedente di operazioni da console non è un elenco completo di tutte le possibili azioni che puoi intraprendere per far sì AWS Management Console che Amazon S3 aggiorni l'oggetto utilizzando la funzionalità di caricamento multiparte. Tieni presente che ogni volta che usi la console per agire su oggetti di dimensioni superiori a 16 MB, il valore del checksum potrebbe non essere il checksum dell'intero oggetto.

Eliminazione di oggetti Amazon S3

Puoi eliminare uno o più oggetti direttamente da Amazon S3 utilizzando la console Amazon S3, gli SDK AWS AWS Command Line Interface ,AWS CLI() o l'API REST. Tutti gli oggetti nel bucket S3 sono soggetti a costi di storage. È pertanto necessario eliminare quelli di cui non si ha più bisogno. Se ad esempio stai eseguendo la raccolta di file di log, è una buona idea eliminarli quando non sono più necessari. È possibile impostare una regola del ciclo di vita per eliminare automaticamente

oggetti come i file di log. Per ulteriori informazioni, consulta the section called “Impostazione della configurazione del ciclo di vita”.

Per informazioni sulle funzionalità e sui prezzi di Amazon S3, consulta Prezzi di Amazon S3.

Quando si elimina un oggetto, sono disponibili le seguenti opzioni API:

- Elimina un singolo oggetto: Amazon S3 fornisce l'API `DELETE` (`DeleteObject`) che consente di eliminare un solo oggetto con una singola richiesta HTTP.
- Elimina più oggetti: Amazon S3 include inoltre l'API per l'eliminazione di più oggetti (`DeleteObjects`) che consente di eliminare fino a 1.000 oggetti con una singola richiesta HTTP.

Quando si eliminano oggetti da un bucket che non è abilitato per le versioni, è necessario specificare solo il nome della chiave oggetto. Tuttavia, quando si eliminano oggetti da un bucket abilitato per le versioni, è possibile specificare facoltativamente l'ID versione dell'oggetto per eliminare una versione specifica dell'oggetto.

Eliminazione a livello di programmazione di oggetti da un bucket abilitato per la versione

Se il bucket è abilitato per le versioni, più versioni dello stesso oggetto possono coesistere nel bucket. Quando si utilizzano i bucket abilitati per le versioni, le operazioni API per l'eliminazione abilitano le opzioni seguenti:

- Specifica una richiesta di eliminazione senza versione: specifichi solo la chiave dell'oggetto senza l'ID versione. In questo caso, Amazon S3 crea un contrassegno di eliminazione e restituisce l'ID versione nella risposta. L'oggetto scompare dal bucket. Per informazioni sulla funzione Controllo delle versioni degli oggetti e sul concetto di contrassegno di eliminazione, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.
- Specifica una richiesta di eliminazione con versione: specifichi sia la chiave dell'oggetto sia l'ID versione. In questo caso sono possibili i due risultati seguenti:
  - Se l'ID versione mappa a una versione dell'oggetto specifica, Amazon S3 elimina la versione specifica dell'oggetto.
  - Se l'ID versione mappa al contrassegno di eliminazione di tale oggetto, Amazon S3 elimina il contrassegno di eliminazione. L'oggetto viene visualizzato nuovamente nel bucket.

Eliminazione di oggetti da un bucket con autenticazione MFA

Quando si eliminano oggetti da un bucket abilitato per l'autenticazione a più fattori (multi-factor authentication, MFA), tieni presente quanto segue:

- Se specifichi un token MFA non valido, la richiesta ha sempre esito negativo.
- Se nel bucket è abilitata l'autenticazione MFA e si effettua una richiesta di eliminazione con versione (si indicano la chiave dell'oggetto e l'ID versione), la richiesta ha esito negativo se non si fornisce un token MFA valido. Inoltre, quando si utilizza l'operazione API per l'eliminazione di più oggetti in un bucket in cui è abilitata l'autenticazione MFA, se una delle eliminazioni è una richiesta di eliminazione con versione (vale a dire, specifica la chiave dell'oggetto e l'ID versione), l'intera richiesta ha esito negativo se non si fornisce un token MFA.

La richiesta ha invece esito positivo nei seguenti casi:

- Se nel bucket è abilitata l'autenticazione MFA ed effettui una richiesta di eliminazione senza versione (non elimini un oggetto con versione) e non fornisci un token MFA, le eliminazioni hanno esito positivo.
- Se una richiesta di eliminazione di più oggetti specifica solo oggetti senza versione da eliminare da un bucket in cui è abilitata l'autenticazione MFA e non fornisci un token MFA, le eliminazioni hanno esito positivo.

Per informazioni sull'eliminazione di MFA, consulta Configurazione dell'eliminazione di MFA.

Argomenti

Eliminazione di un singolo oggetto

Puoi utilizzare la console di Amazon S3 o l'API DELETE per eliminare un singolo oggetto esistente da un bucket S3. Per ulteriori informazioni su come eliminare gli oggetti in Amazon S3, consulta Eliminazione di oggetti Amazon S3 .

Tutti gli oggetti nel bucket S3 sono soggetti a costi di storage. È pertanto necessario eliminare quelli di cui non si ha più bisogno. Se ad esempio si esegue la raccolta di file di log, è una buona idea

eliminarli quando non sono più necessari. È possibile impostare una regola del ciclo di vita per eliminare automaticamente oggetti come i file di log. Per ulteriori informazioni, consulta the section called “Impostazione della configurazione del ciclo di vita”.

Per informazioni sulle funzionalità e sui prezzi di Amazon S3, consulta Prezzi di Amazon S3.

Utilizzo della console S3

Segui questi passaggi per utilizzare la console di Amazon S3 per eliminare un singolo oggetto da un bucket.

> **Warning**
>
> Quando elimini definitivamente un oggetto o una versione dell'oggetto specificata nella console Amazon S3, l'eliminazione non può essere annullata.

Per eliminare un oggetto con il controllo delle versioni abilitato o sospeso

> **Note**
>
> Se l'ID di versione per un oggetto in un bucket con versione sospesa è contrassegnato come`NULL`, S3 elimina definitivamente l'oggetto poiché non esistono versioni precedenti. Tuttavia, se un ID di versione valido è elencato per l'oggetto in un bucket con versione sospesa, S3 crea un marker di eliminazione per l'oggetto eliminato, mantenendo le versioni precedenti dell'oggetto.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket name (Nome bucket) scegliere il nome del bucket dal quale si desidera eliminare un oggetto.
3. Seleziona l'oggetto, quindi scegli Elimina.
4. Per confermare l'eliminazione dell'elenco degli oggetti in Oggetti specificati in Elimina oggetti? casella di testo, immettere**`delete`**.

Per eliminare definitivamente una versione specifica dell'oggetto in un bucket abilitato al controllo delle versioni

> **Warning**
>
> Quando elimini definitivamente una versione specifica dell'oggetto in Amazon S3, l'eliminazione non può essere annullata.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket name (Nome bucket) scegliere il nome del bucket dal quale si desidera eliminare un oggetto.
3. Seleziona l'oggetto da eliminare.
4. Scegli l'interruttore Mostra versioni.
5. Seleziona la versione dell'oggetto, quindi scegli Elimina.
6. Per confermare l'eliminazione permanente delle versioni specifiche degli oggetti elencate in Oggetti specificati, nella sezione Eliminare oggetti? casella di testo, inserisci Elimina definitivamente. Amazon S3 elimina definitivamente la versione specifica dell'oggetto.

Per eliminare definitivamente un oggetto in un bucket Amazon S3 che non ha il controllo delle versioni abilitato

> **Warning**
>
> Quando elimini definitivamente un oggetto in Amazon S3, l'eliminazione non può essere annullata. Inoltre, per tutti i bucket senza il controllo delle versioni abilitato, le eliminazioni sono permanenti.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket name (Nome bucket) scegliere il nome del bucket dal quale si desidera eliminare un oggetto.
3. Seleziona l'oggetto, quindi scegli Elimina.

4. Per confermare l'eliminazione permanente dell'oggetto elencato in Oggetti specificati, nella sezione Eliminare oggetti? casella di testo, inserisci elimina definitivamente.

> **Note**
>
> Se riscontri problemi con l'eliminazione dell'oggetto, consultaVoglio eliminare definitivamente gli oggetti con il controllo delle versioni abilitato.

Utilizzo degli SDK AWS

Gli esempi seguenti mostrano come utilizzare gli AWS SDK per eliminare un oggetto da un bucket. Per ulteriori informazioni, consulta DELETE Object nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Se hai la funzione Controllo delle versioni S3 abilitata sul bucket, sono disponibili le seguenti opzioni:

- Eliminazione di una versione specifica di un oggetto specificando un ID versione.
- Elimina un oggetto senza specificare l'ID versione, nel cui caso Amazon S3 aggiunge un contrassegno di eliminazione all'oggetto.

Per ulteriori informazioni sulla funzione Controllo delle versioni S3, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Java

Example Esempio 1: Eliminazione di un oggetto (bucket senza versione)

Il seguente esempio presuppone che il bucket non disponga della funzione Controllo delle versioni abilitata e che l'oggetto non abbia nessun ID versione. Nella richiesta di eliminazione, si specifica solo la chiave dell'oggetto e non un ID versione.

Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
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 Esempio 2: Eliminazione di un oggetto (bucket con versione)

Nell'esempio seguente viene eliminato un oggetto da un bucket con versione. L'esempio elimina una specifica versione dell'oggetto, specificando il nome della chiave dell'oggetto e l'ID versione.

Inoltre, vengono effettuate le seguenti operazioni:

1. Aggiunge un oggetto campione al bucket. Amazon S3 restituisce l'ID versione del nuovo oggetto aggiunto. L'esempio utilizza questo ID versione nella richiesta di eliminazione.

2. Elimina la versione dell'oggetto, specificando sia il nome della chiave dell'oggetto sia un ID versione. Se non sono disponibili altre versioni dell'oggetto, Amazon S3 elimina l'oggetto interamente. In caso contrario, Amazon S3 elimina solo la versione specificata.

> **Note**
>
> È possibile ottenere gli ID versione di un oggetto inviando una richiesta `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

Gli esempi seguenti mostrano come eliminare un oggetto sia da bucket con versione sia da bucket senza versione. Per ulteriori informazioni sulla funzione Controllo delle versioni S3, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Example Eliminazione di un oggetto da un bucket senza versione

Nell'esempio di codice C# seguente viene eliminato un oggetto da un bucket senza versione. L'esempio presuppone che gli oggetti non dispongano degli ID versione, perciò non vengono specificati gli ID versione. Specifichi solo la chiave dell'oggetto.

Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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 Eliminazione di un oggetto da un bucket con versione

Nell'esempio C# seguente viene eliminato un oggetto da un bucket con versione. Elimina una specifica versione dell'oggetto, specificando il nome della chiave dell'oggetto e l'ID versione.

Il codice esegue le attività sotto elencate:

1. Abilita la funzione Controllo delle versioni S3 su un bucket specificato (se la funzione Versione multiple di S3 è già abilitata, tale operazione non ha effetti).
2. Aggiunge un oggetto campione al bucket. In risposta a questa attività, Amazon S3 restituisce l'ID versione del nuovo oggetto aggiunto. L'esempio utilizza questo ID versione nella richiesta di eliminazione.
3. Elimina l'oggetto campione, specificando sia il nome della chiave dell'oggetto sia un ID versione.

> **Note**
>
> È possibile ottenere l'ID versione di un oggetto anche inviando una richiesta `ListVersions`.
>
> ```
> var listResponse = client.ListVersions(new ListVersionsRequest { BucketName
>  = bucketName, Prefix = keyName });
> ```

Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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

Questo esempio mostra come utilizzare le classi della versione 3 di AWS SDK for PHP per eliminare un oggetto da un bucket senza versione. Per informazioni sull'eliminazione di un oggetto da un bucket con versione, consulta Utilizzo di REST API.

Per ulteriori informazioni sull'API AWS SDK for Ruby, AWS vai a SDK for Ruby - Versione 2.

Nell'esempio PHP seguente viene eliminato un oggetto da un bucket. Poiché questo esempio mostra come eliminare gli oggetti da bucket senza versione, questo fornisce solo il nome bucket e la chiave dell'oggetto (non un ID versione) nella richiesta di eliminazione.

```
<?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();
```

Usando il AWS CLI

Per eliminare un oggetto per richiesta, utilizza l'API `DELETE`. Per ulteriori informazioni, consulta DELETE Object. Per ulteriori informazioni sull'utilizzo della CLI per eliminare un oggetto, consulta la sezione delete-object.

Utilizzo di REST API

È possibile utilizzare gli AWS SDK per eliminare un oggetto. Tuttavia, se l'applicazione lo richiede, è possibile inviare richieste REST direttamente. Per ulteriori informazioni, consulta DELETE Object nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Eliminazione di più oggetti

Tutti gli oggetti nel bucket S3 sono soggetti a costi di storage. È pertanto necessario eliminare quelli di cui non si ha più bisogno. Se ad esempio si esegue la raccolta di file di log, è una buona idea eliminarli quando non sono più necessari. È possibile impostare una regola del ciclo di vita per eliminare automaticamente oggetti come i file di log. Per ulteriori informazioni, consulta the section called “Impostazione della configurazione del ciclo di vita”.

Per informazioni sulle funzionalità e sui prezzi di Amazon S3, consulta Prezzi di Amazon S3.

Puoi utilizzare la console Amazon S3, gli AWS SDK o l'API REST per eliminare più oggetti contemporaneamente da un bucket S3.

Utilizzo della console S3

Segui questi passaggi per utilizzare la console di Amazon S3 per eliminare più oggetti da un bucket.

Warning

- L'eliminazione di un oggetto specificato non può essere annullata.
- Questa azione elimina tutti gli oggetti specificati. Quando si eliminano le cartelle, attendere che l'azione di eliminazione finisca prima di aggiungere nuovi oggetti alla cartella. In caso contrario, potrebbero essere eliminati anche nuovi oggetti.
- Quando si eliminano oggetti in un bucket senza abilitare il controllo delle versioni, Amazon S3 eliminerà definitivamente gli oggetti.
- Quando si eliminano oggetti in un bucket con il controllo delle versioni del bucket abilitato o sospeso, Amazon S3 crea marcatori di eliminazione. Per ulteriori informazioni, consulta Utilizzo dei contrassegni di eliminazione.

Per eliminare oggetti con il controllo delle versioni abilitato o sospeso

Note

Se gli ID di versione per l'oggetto in un bucket con versione sospesa sono contrassegnati come`NULL`, S3 elimina definitivamente gli oggetti poiché non esistono versioni precedenti. Tuttavia, se viene elencato un ID di versione valido per gli oggetti in un bucket con versione sospesa, S3 crea dei marker di eliminazione per gli oggetti eliminati, mantenendo le versioni precedenti degli oggetti.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket name, scegli il nome del bucket da cui desideri eliminare gli oggetti.
3. Seleziona gli oggetti, quindi scegli Elimina.
4. Per confermare l'eliminazione dell'elenco degli oggetti in Oggetti specificati in Elimina oggetti? casella di testo, immettere**`delete`**.

Per eliminare definitivamente versioni di oggetti specifici in un bucket abilitato al controllo delle versioni

> **Warning**
>
> Quando elimini definitivamente versioni di oggetti specifici in Amazon S3, l'eliminazione non può essere annullata.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket name, scegli il nome del bucket da cui desideri eliminare gli oggetti.
3. Selezionare gli oggetti che si intendono eliminare.
4. Scegli l'interruttore Mostra versioni.
5. Seleziona le versioni dell'oggetto, quindi scegli Elimina.
6. Per confermare l'eliminazione permanente delle versioni specifiche degli oggetti elencate in Oggetti specificati, nella sezione Eliminare oggetti? casella di testo, inserisci Elimina definitivamente. Amazon S3 elimina definitivamente le versioni specifiche degli oggetti.

Per eliminare definitivamente gli oggetti in un bucket Amazon S3 per cui il controllo delle versioni non è abilitato

> **Warning**
>
> Quando elimini definitivamente un oggetto in Amazon S3, l'eliminazione non può essere annullata. Inoltre, per tutti i bucket senza il controllo delle versioni abilitato, le eliminazioni sono permanenti.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket name, scegli il nome del bucket da cui desideri eliminare gli oggetti.
3. Seleziona gli oggetti, quindi scegli Elimina.
4. Per confermare l'eliminazione permanente degli oggetti elencati in Oggetti specificati, nella sezione Eliminare oggetti? casella di testo, inserisci elimina definitivamente.

> **Note**
>
> Se riscontri problemi con l'eliminazione degli oggetti, consultaVoglio eliminare definitivamente gli oggetti con il controllo delle versioni abilitato.

Utilizzo degli SDK AWS

Per esempi su come eliminare più oggetti con gli AWS SDK, consulta. Utilizzo `DeleteObjects` con un AWS SDK o una CLI

Per informazioni generali sull'utilizzo di diversi AWS SDK, consulta. Sviluppo con Amazon S3 utilizzando gli SDK AWS

Utilizzo di REST API

Puoi utilizzare gli AWS SDK per eliminare più oggetti utilizzando l'API Multi-Object Delete. Tuttavia, se l'applicazione lo richiede, è possibile inviare richieste REST direttamente.

Per ulteriori informazioni, consulta la sezione relativa all'eliminazione di più oggetti nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Organizzare, elencare e utilizzare gli oggetti

In Amazon S3, puoi utilizzare i prefissi per organizzare lo spazio di storage. Un prefisso è un raggruppamento logico degli oggetti in un bucket. Il valore del prefisso è simile a un nome di directory che consente di archiviare dati simili nella stessa directory in un bucket. Quando si caricano oggetti a livello di programmazione, è possibile utilizzare i prefissi per organizzare i dati.

Nella console di Amazon S3, i prefissi sono chiamati cartelle. È possibile visualizzare tutti gli oggetti e le cartelle nella console S3 passando a un bucket. È inoltre possibile visualizzare informazioni su ciascun oggetto, incluse le proprietà dell'oggetto.

Per ulteriori informazioni sull'elenco e sull'organizzazione dei dati in Amazon S3, consulta i seguenti argomenti.

Argomenti

- Organizzazione degli oggetti utilizzando i prefissi

- Elenco delle chiavi oggetto a livello di programmazione
- Organizzazione degli oggetti nella console di Amazon S3 utilizzando le cartelle
- Visualizzazione della panoramica di un oggetto nella console di Amazon S3
- Visualizzazione delle proprietà di un oggetto nella console di Amazon S3

Organizzazione degli oggetti utilizzando i prefissi

Puoi utilizzare i prefissi per organizzare i dati archiviati nei bucket Amazon S3. Un prefisso è una stringa di caratteri all'inizio del nome della chiave dell'oggetto. Un prefisso può essere di qualsiasi lunghezza, soggetto alla lunghezza massima del nome della chiave dell'oggetto (1.024 byte). Puoi pensare ai prefissi come un modo per organizzare i dati in modo simile alle directory. Tuttavia, i prefissi non sono directory.

La ricerca per prefisso limita i risultati solo alle chiavi che iniziano con il prefisso specificato. Il delimitatore fa in modo che l'operazione di elenco esegua il rollup di tutte le chiavi che condividono un prefisso comune in un unico risultato di elenco di riepilogo

Lo scopo dei parametri Prefix e Delimiter è facilitare l'organizzazione e la visualizzazione delle chiavi in ordine gerarchico. A tale scopo, selezionare un delimitatore per il bucket, ad esempio una barra (/), che non ricorra nei nomi delle chiavi previsti. È possibile utilizzare un altro carattere come delimitatore. Non c'è nulla di unico nel carattere slash (/), ma è un delimitatore di prefisso molto comune. Creare quindi i nomi delle chiavi concatenando tutti i livelli della gerarchia e separando ciascun livello con il delimitatore.

Ad esempio, se si archiviano informazioni sulle città, è possibile organizzarle naturalmente in base al continente, quindi in base al paese, alla provincia o allo stato. Poiché questi nomi in genere non contengono punteggiatura, è possibile selezionare la barra (/) come delimitatore. I seguenti esempi mostrano come utilizzare la barra (/) come delimitatore.

- Europa/Francia/Nouvelle-Aquitaine/Bordeaux
- America del Nord/Canada/Quebec/Montreal
- America del Nord/Stati Uniti/Washington/Bellevue
- America del Nord/Stati Uniti/Washington/Seattle

Se i dati di ogni città del mondo sono stati archiviati in questo modo, sarebbe strano gestire un namespace di chiavi piatto. Utilizzando `Prefix` e `Delimiter` nell'operazione di elenco, puoi

usare la gerarchia creata per elencare i dati. Ad esempio, per elencare tutti gli stati degli Stati Uniti, imposta `Delimiter='/'` e `Prefix='North America/USA/'`. Per elencare tutte le province del Canada per le quali sono disponibili dati, imposta `Delimiter='/'` e `Prefix='North America/Canada/'`.

Per ulteriori informazioni su delimitatori, prefissi e cartelle nidificate, consulta Differenza tra prefissi e cartelle nidificate.

Elenco di oggetti utilizzando prefissi e delimitatori

Se richiedi un elenco con un delimitatore, puoi visualizzare la gerarchia a un solo livello, omettendo e riassumendo le chiavi (possibilmente milioni di esse) nidificate ai livelli più profondi. Ad esempio, supponiamo che tu abbia un bucket (*DOC-EXAMPLE-BUCKET*) con le seguenti chiavi:

`sample.jpg`

`photos/2006/January/sample.jpg`

`photos/2006/February/sample2.jpg`

`photos/2006/February/sample3.jpg`

`photos/2006/February/sample4.jpg`

Il bucket di esempio contiene solo l'oggetto `sample.jpg` a livello root. Per elencare solo gli oggetti a livello root nel bucket, invii una richiesta GET nel bucket con il carattere delimitatore della barra (`/`). In risposta, Amazon S3 restituisce la chiave dell'oggetto `sample.jpg` poiché non contiene il carattere delimitatore `/`. Tutte le altre chiavi contengono questo carattere. Amazon S3 raggruppa queste chiavi e restituisce un singolo elemento `CommonPrefixes` con il valore di prefisso `photos/`, che è una sottostringa dall'inizio di queste chiavi alla prima occorrenza del delimitatore specificato.

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>
```

Per ulteriori informazioni sull'elenco delle chiavi oggetto a livello di programmazione, consulta la sezione Elenco delle chiavi oggetto a livello di programmazione.

Elenco delle chiavi oggetto a livello di programmazione

In Amazon S3, le chiavi possono essere elencate per prefisso. È possibile scegliere un prefisso comune per i nomi delle chiavi correlate e contrassegnare queste chiavi con un carattere speciale che delimita la gerarchia. È quindi possibile utilizzare l'operazione elenco per selezionare e sfogliare le chiavi gerarchicamente. Questa operazione è simile all'archiviazione dei file in directory all'interno di un file system.

Amazon S3 visualizza un'operazione di elenco che consente di elencare le chiavi contenute in un bucket. Le chiavi vengono selezionate per l'elenco in base al bucket e al prefisso. Ad esempio, si prenda in considerazione un bucket denominato "`dictionary`" contenente una chiave per ogni parola inglese. È possibile eseguire una chiamata per elencare tutte le chiavi in tale bucket che iniziano con la lettera "q". I risultati dell'elenco vengono sempre restituiti in ordine binario UTF-8.

Sia le operazioni di elenco SOAP che quelle REST restituiscono un documento XML contenente i nomi delle chiavi corrispondenti e informazioni sull'oggetto identificato da ciascuna chiave.

> **Note**
>
> Il supporto di SOAP su HTTP non viene più utilizzato ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non sono supportate per SOAP. Invece di utilizzare SOAP, ti consigliamo di utilizzare l'API REST o gli AWS SDK.

È possibile raggruppare i gruppi di chiavi che condividono un prefisso che termina con un delimitatore speciale in base al prefisso comune a scopo di elenco. Ciò consente alle applicazioni di organizzare ed esplorare le chiavi in ordine gerarchico, in modo simile all'organizzazione dei file in directory in un file system.

Ad esempio, per estendere il bucket dictionary in modo che contenga altre parole oltre a quelle inglesi, è possibile creare chiavi anteponendo a ciascuna parola un prefisso insieme alla lingua e a un delimitatore, ad esempio "`French/logical`". È possibile utilizzare questo schema di denominazione e la funzione di elenco gerarchico per recuperare un elenco costituito solo dalle parole francesi. È inoltre possibile sfogliare l'elenco di livello superiore delle lingue disponibili senza dover scorrere tutte le chiavi utilizzate in ordine lessicografico. Per ulteriori informazioni su questo tipo di elenco, consulta Organizzazione degli oggetti utilizzando i prefissi.

REST API

Tuttavia, se l'applicazione lo richiede, è possibile inviare richieste REST direttamente. È possibile inviare una richiesta GET per restituire alcuni o tutti gli oggetti in un bucket oppure è possibile utilizzare le policy di selezione per restituire un sottoinsieme degli oggetti in un bucket. Per ulteriori informazioni, consulta l'argomento relativo all'operazione GET Bucket (List Objects) Version 2 nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Efficacia dell'implementazione degli elenchi

Le prestazioni dell'elenco non sono influenzate in modo sostanziale dal numero totale di chiavi nel bucket. Inoltre, non sono influenzate dalla presenza o dall'assenza degli argomenti `delimiter` , `prefix`, `marker` o `maxkeys`.

Scorrimento dei risultati di più pagine

Poiché i bucket possono contenere un numero potenzialmente illimitato di chiavi, una query di elenco può restituire un numero estremamente elevato di risultati. Per gestire set di risultati di grandi dimensioni, l'API di Amazon S3 supporta la paginazione per suddividerli in più risposte. Ciascuna risposta delle chiavi di elenco restituisce una pagina contenente fino a 1000 chiavi con un indicatore che specifica se la risposta è troncata. Invii una serie di richieste di chiavi di elenco finché non ricevi tutte le chiavi. AWS Le librerie wrapper SDK forniscono la stessa impaginazione.

Esempi

I seguenti esempi di codice mostrano come usare. `ListObjects`

CLI

AWS CLI

L'esempio seguente utilizza il `list-objects` comando per visualizzare i nomi di tutti gli oggetti nel bucket specificato:

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size:
 Size}'
```

L'esempio utilizza l'`--query`argomento per filtrare l'output di `list-objects` fino al valore e alla dimensione della chiave per ogni oggetto

Per ulteriori informazioni sugli oggetti, consulta Working with Amazon S3 Objects nella Amazon S3 Developer Guide.

- Per i dettagli sull'API, consulta AWS CLI Command ListObjectsReference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando recupera le informazioni su tutti gli elementi nel bucket «test-files».

```
Get-S3Object -BucketName test-files
```

Esempio 2: questo comando recupera le informazioni sull'elemento "sample.txt" dal bucket «test-files».

```
Get-S3Object -BucketName test-files -Key sample.txt
```

Esempio 3: Questo comando recupera le informazioni su tutti gli elementi con il prefisso «sample» dal bucket «test-files».

```
Get-S3Object -BucketName test-files -KeyPrefix sample
```

- Per i dettagli sull'API, vedere in Cmdlet Reference. ListObjectsAWS Tools for PowerShell

Organizzazione degli oggetti nella console di Amazon S3 utilizzando le cartelle

In Amazon S3 bucket e oggetti sono le risorse primarie e gli oggetti sono archiviati nei bucket. Amazon S3 è caratterizzato da una struttura orizzontale, non da una gerarchia come quella utilizzata in un file system. Tuttavia, per semplicità di organizzazione, la console di Amazon S3 supporta il concetto di cartella come metodo di raggruppamento degli oggetti. La console esegue questa operazione utilizzando un prefisso di nome condiviso per gli oggetti raggruppati. In altre parole, gli oggetti del gruppo hanno nomi che iniziano con una stringa comune. Questa stringa comune, o prefisso condiviso, è il nome della cartella. I nomi degli oggetti sono inoltre noti come nomi chiave.

È possibile, ad esempio, creare una cartella nella console denominata `photos` e memorizzarvi un oggetto denominato `myphoto.jpg`. Tale oggetto viene quindi memorizzato con il nome delle chiave `photos/myphoto.jpg`, di cui `photos/` è il prefisso.

Ecco altri due esempi:

- Se il bucket contiene tre oggetti, `logs/date1.txt`, `logs/date2.txt` e `logs/date3.txt`, la console mostrerà una cartella denominata `logs`. Se si apre la cartella nella console, si vedranno tre oggetti: `date1.txt`, `date2.txt` e `date3.txt`.
- Se si possiede un oggetto denominato `photos/2017/example.jpg`, la console mostrerà una cartella denominata `photos` contenente la cartella `2017`. La cartella `2017` conterrà l'oggetto `example.jpg`.

Si possono avere cartelle nidificate, ma non bucket all'interno di altri bucket. È possibile caricare e copiare gli oggetti direttamente in una cartella. Le cartelle possono essere create, eliminate e rese pubbliche, ma non possono essere rinominate. Gli oggetti possono essere copiati da una cartella all'altra.

> **Important**
>
> Quando crei una cartella in Amazon S3, S3 crea un oggetto con dimensioni pari a 0 byte con una chiave impostata sul nome della cartella fornito. Ad esempio, se crei una cartella denominata `photos` nel bucket, la console di Amazon S3 crea un oggetto della dimensione di 0 byte con la chiave `photos/`. La console crea questo oggetto a supporto del concetto di cartella.

La console di Amazon S3 tratta come cartelle tutti gli oggetti contrassegnati dal carattere di barra (/) come ultimo carattere (quello finale) del nome della chiave (ad esempio `examplekeyname/`). Non è possibile caricare oggetti con un nome chiave che termina con il carattere / mediante la console di Amazon S3. Tuttavia, puoi caricare oggetti denominati con una fine / con l'API Amazon S3 utilizzando AWS Command Line Interface AWS CLI() AWS , SDK o API REST.
Gli oggetti il cui nome termina con / vengono visualizzati come cartelle nella console di Amazon S3. La console di Amazon S3 non mostra il contenuto e i metadati di questi oggetti. Quando si utilizza la console per copiare un oggetto il cui nome termina con /, viene creata una nuova cartella nella posizione di destinazione, ma i dati e i metadati dell'oggetto non vengono copiati.

Argomenti

Creazione di una cartella

Questa sezione spiega come utilizzare la console di Amazon S3 per creare una cartella.

Important

Se la policy del bucket impedisce il caricamento di oggetti in questo bucket senza tag, metadati o assegnatari della lista di controllo degli accessi (ACL), non sarai in grado di creare una cartella utilizzando questa configurazione. Si dovrà invece caricare una cartella vuota e specificare queste impostazioni nella configurazione di caricamento.

Per creare una cartella

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).

3. Nell'elenco Bucket name (Nome bucket) scegliere il nome del bucket in cui si desidera creare una cartella.
4. Se la policy del bucket impedisce il caricamento di oggetti in questo bucket senza crittografia, devi scegliere Enable (Abilita) in Server-side encryption (Crittografia lato server).
5. Scegliere Create folder (Crea cartella).
6. Immettere un nome per la cartella (ad esempio, **`favorite-pics`**). Scegliere Create Folder (Crea cartella).

Creazione di cartelle pubbliche

Consigliamo di bloccare tutto l'accesso pubblico alle cartelle e ai bucket Amazon S3 a meno che non siano necessari una cartella o un bucket pubblici. Quando si rende pubblica una cartella, chiunque su Internet può visualizzare tutti gli oggetti raggruppati nella cartella.

Nella console di Amazon S3 puoi rendere pubblica una cartella. Una cartella può anche essere resa pubblica creando una policy di bucket che ne limita l'accesso ai dati in base al prefisso. Per ulteriori informazioni, consulta Identity and Access Management per Amazon S3.

> **Warning**
>
> Dopo aver reso pubblica una cartella nella console di Amazon S3 non è possibile renderla nuovamente privata. È necessario invece impostare le autorizzazioni per ogni singolo oggetto nella cartella pubblica affinché gli oggetti non abbiano accesso pubblico. Per ulteriori informazioni, consulta Configurazione delle ACL.

Argomenti

- Calcolo delle dimensioni delle cartelle
- Eliminazione di cartelle

Calcolo delle dimensioni delle cartelle

Questa sezione spiega come utilizzare la console di Amazon S3 per calcolare le dimensioni di una cartella.

Calcolo delle dimensioni di una cartella

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket, scegli il nome del bucket in cui è archiviata la cartella.
4. Nell'elenco Objects (Oggetti), seleziona la casella di controllo accanto al nome della cartella.
5. Scegli Actions (Azioni), quindi scegli Calculate total size (Calcola dimensione totale).

> **Note**
>
> Le informazioni sulla cartella (inclusa la dimensione totale) non saranno più disponibili dopo essere usciti dalla pagina. È necessario calcolare nuovamente la dimensione totale se si desidera vederla di nuovo.

> **Important**
>
> - Quando utilizzi l'azione Calculate total size (Calcola dimensione totale) su oggetti o cartelle specificati all'interno del bucket, Amazon S3 calcola il numero totale di oggetti e la dimensione totale dello spazio di archiviazione. Tuttavia, i caricamenti in più parti incompleti o in corso e le versioni precedenti o non correnti non vengono considerati nel calcolo del numero totale di oggetti o della dimensione totale. Questa azione calcola solo il numero totale di oggetti e la dimensione totale per la versione corrente o più recente di ogni oggetto archiviato nel bucket.
>
>   Ad esempio, se nel bucket sono presenti due versioni di un oggetto, il calcolatore dello spazio di archiviazione in Amazon S3 le considera un unico oggetto. Di conseguenza, il numero totale di oggetti calcolato nella console Amazon S3 può differire dalla metrica Object Count mostrata in S3 Storage Lens e dal numero riportato dalla metrica Amazon,. CloudWatch `NumberOfObjects` Allo stesso modo, la dimensione totale dello storage può anche differire dalla metrica Total Storage mostrata in S3 Storage Lens e dalla metrica mostrata in. `BucketSizeBytes` CloudWatch
> - Se il tempo necessario per calcolare la dimensione totale di una cartella di grandi dimensioni impiega troppo tempo, prendi in considerazione l'utilizzo di Amazon S3 Inventory e Amazon S3 Select come alternativa. Innanzitutto, crea una configurazione S3

Inventory per includere i metadati Size per ogni oggetto della cartella grande in un report di inventario. La consegna del primo report di inventario S3 può richiedere fino a 48 ore. Quando il report sull'inventario viene pubblicato, interroga il rapporto di inventario con un'espressione S3 Select `SUM` per aggregare le dimensioni degli oggetti nella cartella. Per ulteriori informazioni, consulta Configurazione dell'inventario utilizzando la console S3 e SUM Esempio.

Eliminazione di cartelle

In questa sezione viene descritto come utilizzare la console di Amazon S3 per eliminare cartelle da un bucket S3.

Per informazioni sulle funzionalità e sui prezzi di Amazon S3, consulta Amazon S3.

Per eliminare cartelle da un bucket S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket name (Nome bucket) scegliere il nome del bucket da cui si desidera eliminare cartelle.
3. Nell'elenco Oggetti, seleziona la casella di controllo accanto alle cartelle e agli oggetti che si desidera eliminare.
4. Scegli Elimina.
5. Nella pagina Delete objects (Elimina oggetti) verifica che siano elencati i nomi delle cartelle selezionate per l'eliminazione.
6. Nella casella Elimina oggetti, immetti **`delete`** e scegli Elimina oggetti.

> **Warning**
>
> Questa azione elimina tutti gli oggetti specificati. Quando si eliminano le cartelle, attendere che l'azione di eliminazione finisca prima di aggiungere nuovi oggetti alla cartella. In caso contrario, potrebbero essere eliminati anche nuovi oggetti.

Visualizzazione della panoramica di un oggetto nella console di Amazon S3

Puoi utilizzare la console di Amazon S3 per visualizzare la panoramica di un oggetto. La console fornisce tutte le informazioni essenziali su un oggetto in un'unica posizione.

Per aprire la pagina dei dettagli di un oggetto

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegliere il nome del bucket contenente l'oggetto.
3. Nell'elenco Oggetti, sceglie il nome dell'oggetto del quale desideri visualizzare la panoramica.

   Sarà visualizzata la pagina dei dettagli dell'oggetto.
4. Per scaricare l'oggetto, scegli Operazioni oggetto, quindi Scarica. Per copiare il percorso dell'oggetto negli appunti, in URL oggetto scegli l'URL.
5. Se nel bucket è abilitata la funzione Controllo delle versioni, scegli Versioni per elencare le versioni dell'oggetto.

   - Per scaricare una versione di un oggetto, seleziona la casella di controllo accanto all'ID versione, scegli Operazioni e quindi Scarica.
   - Per eliminare una versione di un oggetto, seleziona la casella di controllo accanto all'ID versione e scegli Elimina.

> **Important**
>
> È possibile annullare l'eliminazione di un oggetto solo se è stato eliminato come ultima versione (corrente). Non è possibile annullare l'eliminazione della versione precedente di un oggetto eliminato.

Visualizzazione delle proprietà di un oggetto nella console di Amazon S3

Puoi utilizzare la console di Amazon S3 per visualizzare le proprietà di un oggetto, tra cui classe di storage, impostazioni di crittografia, tag e metadati.

Per visualizzare le proprietà di un oggetto

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegliere il nome del bucket contenente l'oggetto.
3. Nell'elenco Oggetti, scegli il nome dell'oggetto del quale desideri visualizzare le proprietà.

   Viene visualizzata la Panoramica dell'oggetto. È possibile scorrere verso il basso per visualizzare le proprietà dell'oggetto.
4. Nella pagina Panoramica dell'oggetto è possibile configurare le proprietà dell'oggetto indicate di seguito.

   > **Note**
   >
   > - Se modifichi una delle proprietà Classe di storage, Crittografia o Metadati, viene creato un nuovo oggetto per sostituire quello precedente. Se è abilitata la funzione Controllo delle versioni S3, viene creata una nuova versione dell'oggetto e l'oggetto esistente diventa una versione precedente. Il ruolo che modifica la proprietà diventa anche il proprietario del nuovo oggetto o della versione dell'oggetto.
   > - Se modifichi le proprietà Storage Class, Encryption o Metadata per un oggetto con tag definiti dall'utente, devi disporre dell'autorizzazione. `s3:GetObjectTagging` Se state modificando queste proprietà per un oggetto che non ha tag definiti dall'utente ma ha una dimensione superiore a 16 MB, è necessario disporre anche dell'autorizzazione. `s3:GetObjectTagging`
   >
   >   Se la policy del bucket di destinazione nega l'`s3:GetObjectTagging`azione, queste proprietà dell'oggetto verranno aggiornate, ma i tag definiti dall'utente verranno rimossi dall'oggetto e verrà visualizzato un errore.

   a. Storage class (Classe di storage): a ogni oggetto in Amazon S3 è associata una classe di storage. La classe di storage che si sceglie di utilizzare dipende dalla frequenza con cui si accede all'oggetto. La classe di storage di default per gli oggetti di S3 è STANDARD. È possibile scegliere la classe di storage quando si carica un oggetto. Per ulteriori informazioni sulle classi di storage, consulta Utilizzo delle classi di storage di Amazon S3.

Per cambiare la classe di storage dopo avere caricato un oggetto, scegliere Storage class (Classe di storage). Scegliere la classe desiderata, quindi selezionare Save (Salva).

b. Impostazioni di crittografia lato server: è possibile utilizzare la crittografia lato server per crittografare gli oggetti S3. Per ulteriori informazioni, consulta le sezioni Specifica della crittografia lato server con AWS KMS (SSE-KMS) o Specifica della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).

c. Metadata (Metadati): ciascun oggetto in Amazon S3 dispone di un set di coppie nome-valore che ne rappresenta i metadati. Per informazioni sull'aggiunta di metadati a un oggetto di S3, consulta Modifica dei metadati degli oggetti nella console di Amazon S3.

d. Tag: è possibile classificare lo storage aggiungendo tag a un oggetto S3. Per ulteriori informazioni, consulta Suddivisione in categorie dello storage utilizzando i tag.

e. Blocco degli oggetti, conservazione e conservazione legali: è possibile impedire l'eliminazione di un oggetto. Per ulteriori informazioni, consulta Utilizzo del blocco oggetti S3.

Utilizzo di URL prefirmati

Per concedere un accesso limitato nel tempo agli oggetti in Amazon S3 senza aggiornare la policy del bucket, puoi utilizzare un URL prefirmato. Un URL prefirmato può essere inserito in un browser o utilizzato da un programma per scaricare un oggetto. Le credenziali utilizzate dall'URL predefinito sono quelle dell' AWS utente che ha generato l'URL.

Puoi anche utilizzare gli URL prefirmati per consentire a qualcuno di caricare un oggetto specifico nel tuo bucket Amazon S3. Ciò consente il caricamento senza richiedere a un'altra parte di disporre di credenziali o autorizzazioni AWS di sicurezza. Se nel bucket esiste già un oggetto con la stessa chiave specificata nell'URL prefirmato, Amazon S3 sostituisce l'oggetto esistente con l'oggetto caricato.

È possibile utilizzare l'URL prefirmato più volte, fino alla data e all'ora di scadenza.

Quando crei un URL prefirmato, devi fornire le credenziali di sicurezza e specificare quanto segue:

- Un bucket Amazon S3
- Una chiave oggetto (se il download di questo oggetto sarà nel tuo bucket Amazon S3, se lo stai caricando questo è il nome del file da caricare)
- Un metodo HTTP (`GET` per scaricare gli oggetti o `PUT` per caricarli)
- Un intervallo di tempo di scadenza

Attualmente, gli URL prefirmati di Amazon S3 non supportano l'utilizzo dei seguenti algoritmi di checksum per l'integrità dei dati (CRC32, CRC32C, SHA-1, SHA-256) quando carichi oggetti. Per verificare l'integrità dell'oggetto dopo il caricamento puoi fornire un digest MD5 dell'oggetto durante il caricamento con un URL prefirmato. Per ulteriori informazioni sull'integrità degli oggetti, consulta Verifica dell'integrità degli oggetti.

Argomenti

Chi può creare un URL prefirmato

Qualsiasi utente che disponga di credenziali di sicurezza valide può creare un URL prefirmato. Tuttavia, per accedere a un oggetto, è necessario che l'URL prefirmato sia creato da un utente che dispone dell'autorizzazione a eseguire l'operazione su cui si basa l'URL prefirmato.

Le credenziali che puoi utilizzare per creare un URL prefirmato sono:

- Profilo dell'istanza IAM: valido fino a 6 ore.
- AWS Security Token Service: valido fino a un massimo di 36 ore se firmato con credenziali di sicurezza a lungo termine o per la durata delle credenziali temporanee, a seconda di quali scadano per prime.
- Utente IAM: valido fino a 7 giorni se utilizzi la versione 4 di AWS Signature.

  Per creare un URL prefirmato valido fino a 7 giorni, devi prima delegare le credenziali dell'utente IAM (la chiave di accesso e la chiave segreta) al metodo in uso per creare l'URL prefirmato.

> **Note**
>
> Se hai creato un URL prefirmato utilizzando credenziali temporanee, l'URL scade insieme alle credenziali. In generale, un URL predefinito scade quando la credenziale utilizzata per crearlo viene revocata, eliminata o disattivata. Ciò avviene anche se l'URL è stato creato con

un orario di scadenza successivo. Per la durata temporanea delle credenziali di sicurezza, consulta AWS STS Comparing API operations nella IAM User Guide.

Tempo di scadenza per gli URL prefirmati

Un URL prefirmato rimane valido per il periodo di tempo specificato al momento della generazione dell'URL. Se crei un URL prefirmato con la console di Amazon S3, il tempo di scadenza può essere impostato tra 1 minuto e 12 ore. Se utilizzi gli AWS SDK AWS CLI o, il tempo di scadenza può essere impostato fino a 7 giorni.

Se hai creato un URL predefinito utilizzando un token temporaneo, l'URL scade alla scadenza del token. In generale, un URL predefinito scade quando la credenziale utilizzata per crearlo viene revocata, eliminata o disattivata. Ciò avviene anche se l'URL è stato creato con un orario di scadenza successivo. Per ulteriori informazioni su come le credenziali utilizzate influiscono sulla data di scadenza, consultaChi può creare un URL prefirmato.

Simple Storage Service (Amazon S3) verifica la data e l'ora di scadenza in un URL firmato al momento della richiesta HTTP. Ad esempio, se un client inizia a scaricare un file di grandi dimensioni immediatamente prima dell'ora di scadenza, il download viene completato anche se l'ora di scadenza viene superata. Se la connessione TCP viene interrotta e il client prova a riavviare il download dopo la scadenza, il download non riesce.

Limitazione delle funzionalità degli URL prefirmati

Le funzionalità dell'URL prefirmato sono limitate dalle autorizzazioni dell'utente che lo ha creato. In sostanza, gli URL prefirmati sono token di connessione che consentono l'accesso agli utenti che li possiedono. Pertanto, consigliamo di proteggerli in modo appropriato. Di seguito sono elencati alcuni metodi che puoi usare per limitare l'uso degli URL prefirmati.

AWS Signature versione 4 (SigV4)

Per applicare un comportamento specifico quando le richieste dell'URL prefirmato vengono autenticate tramite AWS Signature Version 4 (SigV4), puoi utilizzare le chiavi di condizione nelle policy del bucket e nelle policy dei punti di accesso. Ad esempio, la policy del bucket seguente utilizza la condizione `s3:signatureAge` per negare qualsiasi richiesta di URL prefirmato da Amazon S3 sugli oggetti nel bucket *`example-s3-bucket1`* se la firma ha più di 10 minuti. Per utilizzare questo comando, sostituisci *`user input placeholders`* con le tue informazioni.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Deny a presigned URL request if the signature is more than 10 min
 old",
         "Effect": "Deny",
         "Principal": {"AWS":"*"},
         "Action": "s3:*",
         "Resource": "arn:aws:s3:::example-s3-bucket1/*",
         "Condition": {
            "NumericGreaterThan": {
               "s3:signatureAge": 600000
            }
         }
      }
   ]
}
```

Per ulteriori informazioni sulla versione 4 di AWS Signature relativa alle chiavi di policy, consulta AWS Signature Version 4 Authentication nel riferimento all'API di Amazon Simple Storage Service.

Limitazioni per percorso di rete

Se desideri limitare l'uso di URL predefiniti e tutti gli accessi di Amazon S3 a determinati percorsi di rete, puoi AWS Identity and Access Management scrivere policy (IAM). Puoi impostare queste policy sul principale del servizio IAM che effettua la chiamata, sul bucket Amazon S3 o su entrambi.

Una restrizione del percorso di rete sul principale IAM richiede all'utente di tali credenziali di effettuare le richieste dalla rete specificata. Una restrizione sul bucket o sul punto di accesso richiede che tutte le richieste a quella risorsa provengano dalla rete specificata. Queste restrizioni si applicano anche al di fuori dello scenario di URL prefirmato.

La chiave della condizione globale IAM utilizzata dipende dal tipo di endpoint. Se utilizzi l'endpoint pubblico per Amazon S3, utilizza `aws:SourceIp`. Se utilizzi un endpoint di cloud privato virtuale (VPC) per Amazon S3, usa `aws:SourceVpc` o `aws:SourceVpce`.

La seguente dichiarazione sulla politica IAM richiede che il principale acceda AWS solo dall'intervallo di rete specificato. Con questa istruzione della policy, tutti gli accessi devono avere origine da tale intervallo. Ciò include il caso di un utente che utilizza un URL prefirmato per Amazon S3. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
{
    "Sid": "NetworkRestrictionForIAMPrincipal",
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
        "NotIpAddressIfExists": {"aws:SourceIp": "IP-address-range"},
        "BoolIfExists": {"aws:ViaAWSService": "false"}
    }
}
```

Per ulteriori esempi di policy di bucket che utilizzano la chiave `aws:SourceIp` AWS global condition per limitare l'accesso a un bucket Amazon S3 a un intervallo di rete specifico, consulta. Gestione dell'accesso in base a indirizzi IP specifici

Condivisione di oggetti mediante URL prefirmati

Per impostazione predefinita, tutti gli oggetti di Amazon S3 sono privati, solo il proprietario dell'oggetto dispone dell'autorizzazione per accedere agli oggetti di Amazon S3. Tuttavia, il proprietario dell'oggetto può condividere oggetti con altri creando un URL prefirmato. Un URL prefirmato utilizza le credenziali di sicurezza per concedere un'autorizzazione limitata nel tempo per scaricare oggetti. L'URL può essere inserito in un browser o utilizzato da un programma per scaricare l'oggetto. Le credenziali utilizzate dall'URL predefinito sono quelle dell' AWS utente che ha generato l'URL.

Per ulteriori informazioni sugli URL prefirmati, consulta Utilizzo di URL prefirmati.

Puoi creare un URL predefinito per condividere un oggetto senza scrivere alcun codice utilizzando la console Amazon S3 AWS , Explorer for Visual Studio (Windows) o. AWS Toolkit for Visual Studio Code Puoi anche generare un URL predefinito a livello di codice utilizzando AWS Command Line Interface ()AWS CLI o gli SDK. AWS

Utilizzo della console S3

Puoi utilizzare la console Amazon S3 per generare un URL prefirmato per un oggetto seguendo questi fasi. Nella console, il tempo massimo di scadenza per un URL prefirmato è di 12 ore dal momento della creazione.

Generazione di un URL prefirmato utilizzando la console di Amazon S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckets (Bucket) scegli il nome del bucket contenente gli oggetto per cui desideri ottenere l'URL prefirmato.
4. Nell'elenco Objects (Oggetti), seleziona l'oggetto per cui desideri creare un URL prefirmato.
5. Nel menu Operazioni oggetti, scegli Crea URL prefirmato.
6. Specifica per quanto tempo desideri che l'URL prefirmato sia valido.
7. Scegli Create presigned URL (Crea URL prefirmato).
8. Quando viene visualizzata la conferma, l'URL viene automaticamente copiato negli appunti. Verrà visualizzato un pulsante per copiare l'URL preimpostato qualora fosse necessario copiarlo di nuovo.

Usando il AWS CLI

Il seguente AWS CLI comando di esempio genera un URL predefinito per la condivisione di un oggetto da un bucket Amazon S3. Quando utilizzi il AWS CLI, il tempo di scadenza massimo per un URL predefinito è di 7 giorni dal momento della creazione. Per utilizzare questo comando, sostituisci *`user input placeholders`* con le tue informazioni.

```
aws s3 presign s3://example-s3-bucket1/mydoc.txt --expires-in 604800
```

> **Note**
>
> Per tutti Regioni AWS quelli lanciati dopo il 20 marzo 2019 è necessario specificare l'`endpoint-urle` `Regione` `AWS` con la richiesta. Per un elenco degli endpoint e delle regioni Amazon S3 disponibili, consulta Regioni ed endpoint in Riferimenti generali AWS .

```
aws s3 presign s3://example-s3-bucket1/mydoc.txt --expires-in 604800 --region af-south-1 --endpoint-url https://s3.af-south-1.amazonaws.com
```

Per ulteriori informazioni, consulta la sezione presign nella Documentazione di riferimento della AWS CLI .

Utilizzo degli AWS SDK

Per esempi di utilizzo degli AWS SDK per generare un URL predefinito per la condivisione di un oggetto, consulta Creare un URL predefinito per Amazon S3 utilizzando un SDK. AWS

Quando utilizzi gli AWS SDK per generare un URL predefinito, il tempo di scadenza massimo è di 7 giorni dal momento della creazione.

> **Note**
>
> Per tutti Regioni AWS quelli lanciati dopo il 20 marzo 2019 è necessario specificare l'`endpoint-url` `Regione AWS` con la richiesta. Per un elenco degli endpoint e delle regioni Amazon S3 disponibili, consulta Regioni ed endpoint in Riferimenti generali AWS .

> **Note**
>
> Quando si utilizzano gli AWS SDK, l'attributo Tagging deve essere un'intestazione e non un parametro di query. Tutti gli altri attributi possono essere passati come parametri per l'URL prefirmato.

Utilizzo di (Windows AWS Toolkit for Visual Studio)

> **Note**
>
> Al momento, non AWS Toolkit for Visual Studio supporta Visual Studio per Mac.

1. Installa AWS Toolkit for Visual Studio utilizzando le seguenti istruzioni, Installazione e configurazione del Toolkit for Visual Studio nella Guida per AWS Toolkit for Visual Studio l'utente.
2. Effettuare la connessione AWS utilizzando i seguenti passaggi, Connessione a AWS nella Guida per AWS Toolkit for Visual Studio l'utente.
3. Nel pannello laterale sinistro denominato AWS Explorer, fai doppio clic sul bucket contenente l'oggetto.

4. Fai clic con il pulsante destro del mouse sull'oggetto per cui desideri che venga generato un URL predefinito e seleziona Crea URL prefirmato... .
5. Nella finestra pop-up, imposta la data e l'ora di scadenza dell'URL predefinito.
6. La chiave dell'oggetto dovrebbe essere precompilata in base all'oggetto selezionato.
7. Scegli GET per specificare che questo URL prefirmato verrà utilizzato per scaricare un oggetto.
8. Scegli il pulsante Genera.
9. Per copiare l'URL negli appunti, scegliere Copia.
10. Per utilizzare l'URL predefinito generato, incollate l'URL in qualsiasi browser.

Usando AWS Toolkit for Visual Studio Code

Se utilizzi Visual Studio Code, puoi generare un URL prefirmato per condividere un oggetto senza scrivere codice tramite AWS Toolkit for Visual Studio Code. Per ulteriori informazioni, consulta AWS Toolkit for Visual Studio Code nella Guida per l'utente di AWS Toolkit for Visual Studio Code .

Per istruzioni su come installare AWS Toolkit for Visual Studio Code, vedere Installazione di AWS Toolkit for Visual Studio Code nella Guida per l'AWS Toolkit for Visual Studio Code utente.

1. Effettuare la connessione AWS utilizzando i seguenti passaggi, Connessione a AWS Toolkit for Visual Studio Code nella Guida per AWS Toolkit for Visual Studio Code l'utente.
2. Seleziona il AWS logo nel pannello di sinistra in Visual Studio Code.
3. In EXPLORER, seleziona S3.
4. Scegli un bucket e un file e apri il menu contestuale (tasto destro del mouse).
5. Scegli Genera URL prefirmato, quindi imposta l'ora di scadenza (in minuti).
6. Premi Invio e l'URL prefirmato verrà copiato negli appunti.

Caricamento di oggetti con URL prefirmati

Puoi utilizzare gli URL prefirmati per consentire a qualcuno di caricare un oggetto nel tuo bucket Amazon S3. L'utilizzo di un URL predefinito consentirà il caricamento senza richiedere a terzi di disporre di credenziali o autorizzazioni AWS di sicurezza. Le funzionalità dell'URL prefirmato sono limitate dalle autorizzazioni dell'utente che lo ha creato. In altre parole, se si riceve un URL prefirmato per caricare un oggetto, è possibile caricarlo solo se il creatore dell'URL dispone delle autorizzazioni necessarie per caricare tale oggetto.

Quando carichi un oggetto nel bucket utilizzando l'URL, Amazon S3 crea l'oggetto in un bucket specifico. Se nel bucket esiste già un oggetto con la stessa chiave specificata nell'URL prefirmato, Amazon S3 sostituisce l'oggetto esistente con l'oggetto caricato. Dopo il caricamento, il proprietario del bucket sarà il proprietario dell'oggetto.

Per ulteriori informazioni sugli URL prefirmati, consulta Utilizzo di URL prefirmati.

Genera un URL prefirmato per un oggetto senza scrivere alcun codice mediante AWS Explorer per Visual Studio. È possibile generare un URL prefirmato in modo programmatico utilizzando AWS SDK.

Utilizzo di (Windows AWS Toolkit for Visual Studio)

> **Note**
>
> Al momento, non AWS Toolkit for Visual Studio supporta Visual Studio per Mac.

1. Installa AWS Toolkit for Visual Studio utilizzando le seguenti istruzioni, Installazione e configurazione del Toolkit for Visual Studio nella Guida per AWS Toolkit for Visual Studio l'utente.
2. Effettuare la connessione AWS utilizzando i seguenti passaggi, Connessione a AWS nella Guida per AWS Toolkit for Visual Studio l'utente.
3. Nel pannello laterale sinistro denominato AWS Explorer, fai clic con il pulsante destro del mouse sul bucket in cui desideri caricare un oggetto.
4. Scegli Crea URL prefirmato... .
5. Nella finestra pop-up, imposta la data e l'ora di scadenza dell'URL predefinito.
6. Per Object Key, imposta il nome del file da caricare. Il file che stai caricando deve corrispondere esattamente a questo nome. Se nel bucket esiste già un oggetto con la stessa chiave oggetto, Amazon S3 sostituirà l'oggetto esistente con l'oggetto appena caricato.
7. Scegli PUT per specificare che questo URL prefirmato verrà utilizzato per caricare un oggetto.
8. Scegli il pulsante Genera.
9. Per copiare l'URL negli appunti, scegliere Copia.
10. Per utilizzare questo URL, puoi inviare una richiesta PUT con il comando `curl`. Includi il percorso completo del file e l'URL predefinito stesso.

```
curl -X PUT -T "/path/to/file" "presigned URL"
```

Utilizzo degli SDK AWS

Per esempi di utilizzo degli AWS SDK per generare un URL predefinito per il caricamento di un oggetto, consulta Creare un URL predefinito per Amazon S3 utilizzando un SDK. AWS

Quando utilizzi gli AWS SDK per generare un URL predefinito, il tempo di scadenza massimo è di 7 giorni dal momento della creazione.

> **Note**
>
> Per tutti Regioni AWS quelli lanciati dopo il 20 marzo 2019 è necessario specificare l'`endpoint-url`e `Regione AWS` con la richiesta. Per un elenco degli endpoint e delle regioni Amazon S3 disponibili, consulta Regioni ed endpoint in Riferimenti generali AWS .

Trasformazione di oggetti con S3 Object Lambda

Con Amazon S3 Object Lambda puoi aggiungere codice personalizzato alle richieste `GET`, `LIST` e `HEAD` di Amazon S3 per modificare ed elaborare i dati restituiti a un'applicazione. Puoi utilizzare il codice personalizzato per modificare i dati restituiti dalle richieste `GET` S3 standard per filtrare le righe, ridimensionare e applicare la filigrana alle immagini in modo dinamico, oscurare i dati riservati e molto altro. Puoi anche utilizzare S3 Object Lambda per modificare l'output delle richieste `LIST` S3 per creare una vista personalizzata di tutti gli oggetti in un bucket e delle richieste `HEAD` S3 per modificare i metadati degli oggetti come il nome e la dimensione dell'oggetto. Puoi utilizzare S3 Object Lambda come origine per la tua distribuzione CloudFront Amazon per personalizzare i dati per gli utenti finali, ad esempio ridimensionando automaticamente le immagini, transcodificando formati più vecchi (come da JPEG a WebP) o rimuovendo i metadati. Per ulteriori informazioni, consulta il post del AWS blog Usa Amazon S3 Object Lambda con Amazon. CloudFront Basato sulle funzioni AWS Lambda, il codice viene eseguito su un'infrastruttura completamente gestita da. AWS L'utilizzo di S3 Object Lambda riduce la necessità di creare e archiviare copie derivate dei dati o di eseguire proxy senza dover modificare le applicazioni.

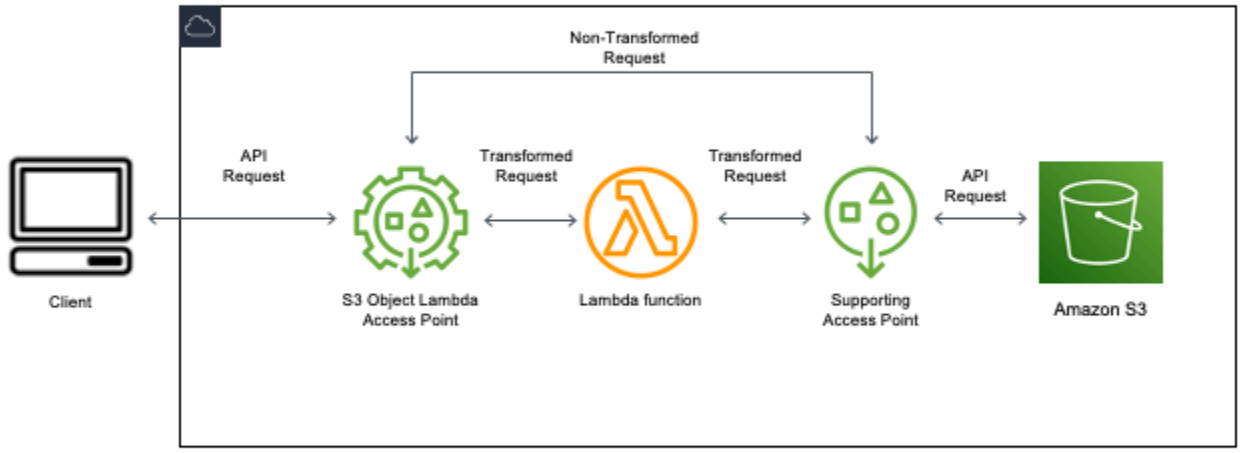
Come funziona S3 Object Lambda

S3 Object Lambda AWS Lambda utilizza funzioni per elaborare automaticamente l'output di `GET` S3 o richieste `LIST` standard. `HEAD` AWS Lambda è un servizio di elaborazione serverless che esegue codice definito dal cliente senza richiedere la gestione delle risorse di elaborazione sottostanti. Puoi

creare ed eseguire le funzioni Lambda personalizzate, adattando la trasformazione dei dati a casi d'uso specifici.

Dopo averla configurata, puoi collegare una funzione Lambda a un endpoint del servizio Lambda per oggetti S3, noto come punto di accesso Lambda per oggetti. Il punto di accesso Lambda per oggetti utilizza un punto di accesso S3 standard, noto come punto di accesso di supporto, per accedere ad Amazon S3.

Quando invii una richiesta al punto di accesso Lambda per oggetti, Amazon S3 richiama automaticamente la funzione Lambda. Tutti i dati recuperati utilizzando una richiesta S3 `GET`, `LIST` o `HEAD` tramite il punto di accesso Lambda per oggetti restituirà un risultato trasformato all'applicazione. Tutte le altre richieste vengono elaborate normalmente, come illustrato nel diagramma seguente.



Gli argomenti in questa sezione descrivono come utilizzare S3 Object Lambda.

Argomenti

- Creazione di punti di accesso Object Lambda
- Utilizzo dei punti di accesso Amazon S3 Object Lambda
- Considerazioni sulla sicurezza per i punti di accesso S3 Object Lambda

- Scrittura di funzioni Lambda per i punti di accesso Lambda per oggetti S3
- Utilizzo delle AWS funzioni Lambda integrate
- Best practice e linee guida per S3 Object Lambda
- Tutorial di S3 Object Lambda
- Debug di S3 Object Lambda

Creazione di punti di accesso Object Lambda

Un punto di accesso Lambda per oggetti è associato esattamente a un punto di accesso standard e quindi a un bucket Amazon S3. Per creare un punto di accesso Lambda per oggetti, sono necessarie le seguenti risorse:

- Un bucket Amazon S3. Per ulteriori informazioni sulla creazione dei bucket, consulta la sezione the section called “Creazione di un bucket”.
- Un punto di accesso S3 standard. Quando utilizzi i punti di accesso Lambda per oggetti, questo punto di accesso standard è noto come punto di accesso di supporto. Per informazioni sulla creazione di punti di accesso standard, consulta la sezione the section called “Creazione di access point”.
- Una funzione. AWS Lambda Puoi creare una funzione Lambda personalizzata oppure utilizzare una funzione predefinita. Per ulteriori informazioni sulla creazione delle funzioni Lambda, consulta the section called “Scrittura delle funzioni Lambda”. Per ulteriori informazioni sulle funzioni incorporate, consulta Utilizzo delle AWS funzioni Lambda integrate.
- (Facoltativo) Una politica AWS Identity and Access Management (IAM). I punti di accesso Amazon S3 supportano le policy delle risorse IAM che consentono di controllare l'utilizzo del punto di accesso per risorsa, utente o altre condizioni. Per ulteriori informazioni sulla creazione di queste policy, consulta the section called “Configurazione delle policy IAM”.

Nelle sezioni seguenti viene descritto come creare un punto di accesso Lambda per oggetti utilizzando:

- La AWS Management Console
- Il AWS Command Line Interface (AWS CLI)
- Un AWS CloudFormation modello
- Il AWS Cloud Development Kit (AWS CDK)

Per ulteriori informazioni su come creare un punto di accesso Lambda per oggetti tramite REST API, consulta `CreateAccessPointForObjectLambda` nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Creazione di un punto di accesso Lambda per oggetti

Utilizza una delle procedure riportate di seguito per creare il punto di accesso Lambda per oggetti.

Utilizzo della console S3

Per creare un punto di accesso Lambda per oggetti utilizzando la console

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nella barra di navigazione, scegli il nome del file attualmente visualizzato Regione AWS. Quindi, scegli la regione a cui vuoi passare.
3. Nel pannello di navigazione a sinistra, scegli Object Lambda Access Points (Punti di accesso Object Lambda).
4. Nella pagina Object Lambda Access Points (Punti di accesso Object Lambda), scegli Create Object Lambda Access Point (Crea punto di accesso Object Lambda).
5. In Object Lambda Access Point name (Nome punto di accesso Object Lambda), specifica il nome da utilizzare per il punto di accesso.

   Come per i punti di accesso standard, esistono regole per la denominazione dei punti di accesso Lambda per oggetti. Per ulteriori informazioni, consulta Regole per la denominazione degli Punti di accesso Amazon S3.
6. In Supporting Access Point (Punto di accesso di supporto), specifica o seleziona il punto di accesso standard da utilizzare. Il punto di accesso deve trovarsi nella Regione AWS stessa posizione degli oggetti che desiderate trasformare. Per informazioni sulla creazione di punti di accesso standard, consulta la sezione the section called “Creazione di access point”.
7. In Configurazione della trasformazione puoi aggiungere una funzione che trasforma i dati per il punto di accesso Lambda per oggetti. Esegui una di queste operazioni:

   - Se hai già una AWS Lambda funzione nel tuo account, puoi sceglierla nella funzione Invoke Lambda. Qui puoi inserire l'Amazon Resource Name (ARN) di una funzione Lambda nel tuo o Account AWS scegliere una funzione Lambda dal menu a discesa.
   - Se desideri utilizzare una funzione AWS incorporata, scegli il nome della funzione in funzione AWS integrata e seleziona Crea funzione Lambda. Verrai indirizzato alla console

Lambda dove potrai implementare una funzione integrata nel tuo. Account AWS Per ulteriori informazioni sulle funzioni integrate, consulta Utilizzo delle AWS funzioni Lambda integrate.

In S3 APIs (API S3), scegli una o più operazioni API da richiamare. Per ogni API selezionata, devi specificare una funzione Lambda da richiamare.

8. (Facoltativo) In Payload, aggiungi il testo JSON da fornire come input alla funzione Lambda. Puoi configurare payload con parametri diversi per diversi punti di accesso Lambda per oggetti che richiamano la stessa funzione Lambda, estendendo così la flessibilità della funzione stessa.

   > **Important**
   >
   > Quando utilizzi i punti di accesso Lambda per oggetti, assicurati che il payload non contenga informazioni riservate.

9. (Facoltativo) In Range and part number (Intervallo e numero parte), devi abilitare l'opzione se vuoi elaborare le richieste `GET` e `HEAD` con intestazioni di intervallo e numero di parte. Selezionando questa opzione confermi che la funzione Lambda è in grado di riconoscere ed elaborare queste richieste. Per ulteriori informazioni sulle intestazioni di intervalli e numeri di parte, consulta Utilizzo delle intestazioni Range e partNumber.
10. (Facoltativo) In Parametri di richiesta seleziona Abilita o Disabilita per aggiungere il monitoraggio Amazon S3 al punto di accesso Lambda per oggetti. Le metriche delle richieste vengono fatturate alla tariffa standard di Amazon CloudWatch.
11. (Facoltativo) In Object Lambda Access Point policy (Policy punto di accesso Object Lambda), imposta una policy delle risorse. Le policy delle risorse concedono le autorizzazioni per il punto di accesso Lambda per oggetti specificato e possono controllare l'utilizzo del punto di accesso per risorsa, utente o altre condizioni. Per ulteriori informazioni sulle policy delle risorse per i punti di accesso Object Lambda, consulta Configurazione delle policy IAM per i punti di accesso Lambda per oggetti.
12. In Block Public Access settings for this Object Lambda Access Point (Impostazioni blocco accesso pubblico per questo punto di accesso Object Lambda), seleziona le impostazioni di Blocco dell'accesso pubblico Amazon S3 da applicare al punto di accesso. Tutte le impostazioni di blocco dell'accesso pubblico sono abilitate per impostazione predefinita per i nuovi punti di accesso Object Lambda. È consigliabile non modificare questa impostazione predefinita. Amazon S3 attualmente non supporta la modifica delle impostazioni di blocco dell'accesso pubblico dei punti di accesso dopo la creazione del punto di accesso Object Lambda.

Per ulteriori informazioni sull'utilizzo del blocco dell'accesso pubblico in Amazon S3, consulta Gestione dell'accesso pubblico agli access point.

13. Seleziona Create Object Lambda Access Point (Crea punto di accesso Object Lambda).

Utilizzando il AWS CLI

Per creare un punto di accesso Object Lambda utilizzando un modello AWS CloudFormation

> **Note**
>
> Per utilizzare i seguenti comandi, sostituisci *`user input placeholders`* con le tue specifiche informazioni.

1. Scarica il pacchetto di distribuzione delle AWS Lambda funzioni `s3objectlambda_deployment_package.zip` nella configurazione predefinita di S3 Object Lambda.
2. Esegui il seguente comando `put-object` per caricare il pacchetto in un bucket Amazon S3.

```
aws s3api put-object --bucket Amazon S3 bucket name --key
 s3objectlambda_deployment_package.zip --body release/
s3objectlambda_deployment_package.zip
```

3. Scarica il AWS CloudFormation modello nella configurazione `s3objectlambda_defaultconfig.yaml` predefinita di S3 Object Lambda.
4. Esegui il seguente comando `deploy` per implementare il modello nell' Account AWS in uso.

```
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
```

Puoi configurare questo AWS CloudFormation modello per richiamare le operazioni `GET` Lambda `HEAD` `LIST` e API. Per ulteriori informazioni sulla modifica della configurazione predefinita del modello, consulta the section called “Automatizza la configurazione di S3 Object Lambda con AWS CloudFormation”.

Per creare un Object Lambda Access Point utilizzando AWS CLI

> **Note**
>
> Per utilizzare i seguenti comandi, sostituisci *`user input placeholders`* con le tue specifiche informazioni.

Nell'esempio seguente viene creato un punto di accesso Lambda per oggetti denominato *`my-object-lambda-ap`* per il bucket *`DOC-EXAMPLE-BUCKET1`* nell'account *`111122223333`*. L'esempio presuppone che sia già stato creato un punto di accesso standard denominato *`example-ap`*. Per informazioni sulla creazione di un punto di accesso standard, consulta la sezione the section called “Creazione di access point”.

Questo esempio utilizza la funzione AWS predefinita. `decompress` Per ulteriori informazioni sulle funzioni incorporate, consulta the section called “Utilizzo di funzioni AWS integrate”.

1. Creare un bucket. In questo esempio verrà utilizzato *`DOC-EXAMPLE-BUCKET1`*. Per ulteriori informazioni sulla creazione dei bucket, consulta la sezione the section called “Creazione di un bucket”.
2. Creare un punto di accesso standard e collegarlo al bucket. In questo esempio verrà utilizzato *`example-ap`*. Per informazioni sulla creazione di punti di accesso standard, consulta la sezione the section called “Creazione di access point”.
3. Esegui una di queste operazioni:

   - Crea una funzione Lambda nel tuo account da utilizzare per trasformare l'oggetto Amazon S3. Per ulteriori informazioni sulla creazione delle funzioni Lambda, consulta the section called “Scrittura delle funzioni Lambda”. Per utilizzare la funzione personalizzata con AWS CLI, consulta Using Lambda with the AWS CLI nella AWS Lambda Developer Guide.
   - Usa una funzione AWS Lambda predefinita. Per ulteriori informazioni sulle funzioni incorporate, consulta Utilizzo delle AWS funzioni Lambda integrate.
4. Crea un file di configurazione JSON denominato `my-olap-configuration.json`. In questa configurazione, fornisci il punto di accesso di supporto e il nome della risorsa Amazon (ARN) per

la funzione Lambda creata nei passaggi precedenti o l'ARN per la funzione predefinita che stai utilizzando.

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. Esegui il comando `create-access-point-for-object-lambda` per creare il punto di accesso Lambda per oggetti.

```
aws s3control create-access-point-for-object-lambda --account-id 111122223333 --
name my-object-lambda-ap --configuration file://my-olap-configuration.json
```

6. (Facoltativo) Crea un file di policy JSON denominato `my-olap-policy.json`.

   L'aggiunta di una policy delle risorse per il punto di accesso Object Lambda può controllare l'utilizzo del punto di accesso per risorsa, utente o altre condizioni. Questa policy delle risorse concede l'autorizzazione `GetObject` per l'account *444455556666* al punto di accesso Lambda per oggetti specificato.

   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. (Facoltativo) Esegui il comando `put-access-point-policy-for-object-lambda` per impostare la policy delle risorse.

```
aws s3control put-access-point-policy-for-object-lambda --account-id 111122223333
 --name my-object-lambda-ap --policy file://my-olap-policy.json
```

8. (Facoltativo) Specifica un payload.

   Un payload è un file JSON opzionale che puoi fornire alla tua AWS Lambda funzione come input. Puoi configurare payload con parametri diversi per diversi punti di accesso Lambda per oggetti che richiamano la stessa funzione Lambda, estendendo così la flessibilità della funzione stessa.

   La seguente configurazione del punto di accesso Lambda per oggetti mostra un payload con due parametri.

```
{
 "SupportingAccessPoint": "AccessPointArn",
 "CloudWatchMetricsEnabled": false,
 "TransformationConfigurations": [{
  "Actions": ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
  "ContentTransformation": {
   "AwsLambda": {
    "FunctionArn": "FunctionArn",
    "FunctionPayload": "{\"res-x\": \"100\",\"res-y\": \"100\"}"
   }
  }
 }]
}
```

   La seguente configurazione del punto di accesso Lambda per oggetti mostra un payload con un parametro e con `GetObject-Range`, `GetObject-PartNumber`, `HeadObject-Range` e `HeadObject-PartNumber` abilitati.

```
{
    "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**
>
> Quando utilizzi i punti di accesso Lambda per oggetti, assicurati che il payload non contenga informazioni riservate.

Utilizzo della AWS CloudFormation console e del modello

È possibile creare un punto di accesso Lambda per oggetti utilizzando la configurazione predefinita fornita da Amazon S3. Puoi scaricare un AWS CloudFormation modello e il codice sorgente della funzione Lambda dal GitHub repository e distribuire queste risorse per configurare un Object Lambda Access Point funzionale.

Per informazioni sulla modifica della configurazione predefinita del AWS CloudFormation modello, consulta. the section called “Automatizza la configurazione di S3 Object Lambda con AWS CloudFormation”

Per informazioni sulla configurazione degli access point Object Lambda AWS CloudFormation utilizzando senza il modello, `AWS::S3ObjectLambda::AccessPoint`consultate la Guida per AWS CloudFormation l'utente.

Per caricare il pacchetto di implementazione della funzione Lambda

1. Scarica il pacchetto di distribuzione delle AWS Lambda funzioni `s3objectlambda_deployment_package.zip` nella configurazione predefinita di S3 Object Lambda.
2. Carica il pacchetto in un bucket Amazon S3

Per creare un punto di accesso Object Lambda utilizzando la console AWS CloudFormation

1. Scarica il AWS CloudFormation modello nella configurazione `s3objectlambda_defaultconfig.yaml` predefinita di S3 Object Lambda.
2. Accedi alla console di AWS gestione e apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
3. Esegui una di queste operazioni:
   - Se non l'hai mai usato AWS CloudFormation prima, nella AWS CloudFormation home page, scegli Crea stack.
   - Se l'hai AWS CloudFormation già usato, nel riquadro di navigazione a sinistra, scegli Stacks. Scegli Create stack (Crea stack), quindi With new resources (standard) Con nuove risorse (standard).
4. Per Prerequisito - Prepara modello, scegliere Il modello è pronto.
5. In Specify template (Specifica modello), scegli Upload a template file (Carica un file modello), quindi carica `s3objectlambda_defaultconfig.yaml`.
6. Seleziona Successivo.
7. Nella pagina Specifica dettagli della pila, immetti un nome per la pila.
8. Nella sezione Parameters (Parametri), specifica i seguenti parametri definiti nel modello dello stack:
   a. Per CreateNewSupportingAccessPoint, esegui una delle seguenti operazioni:
      - Se disponi già di un punto di accesso di supporto per il bucket S3 in cui hai caricato il modello, scegli false.
      - Se intendi creare un nuovo punto di accesso per questo bucket, scegli true.
   b. Infatti EnableCloudWatchMonitoring, scegli true o false, a seconda che tu voglia abilitare i parametri e gli allarmi delle CloudWatch richieste Amazon.

c. (Facoltativo) Per LambdaFunctionPayload, aggiungi il testo JSON che desideri fornire alla tua funzione Lambda come input. Puoi configurare payload con parametri diversi per diversi punti di accesso Lambda per oggetti che richiamano la stessa funzione Lambda, estendendo così la flessibilità della funzione stessa.

> **Important**
>
> Quando utilizzi i punti di accesso Lambda per oggetti, assicurati che il payload non contenga informazioni riservate.

d. Per LambdaFunctionRuntime, inserisci il tuo runtime preferito per la funzione Lambda. Le scelte disponibili sono `nodejs14.x`, `python3.9`, `java11`.

e. Per LambdaFunctionS3 BucketName, inserisci il nome del bucket Amazon S3 in cui hai caricato il pacchetto di distribuzione.

f. Per LambdaFunctionS3Key, inserisci la chiave oggetto Amazon S3 in cui hai caricato il pacchetto di distribuzione.

g. Per LambdaFunctionS3 ObjectVersion, inserisci la versione dell'oggetto Amazon S3 in cui hai caricato il pacchetto di distribuzione.

h. Per ObjectLambdaAccessPointName, inserisci un nome per il tuo Object Lambda Access Point.

i. Per S3 BucketName, inserisci il nome del bucket Amazon S3 che verrà associato al tuo access point Object Lambda.

j. Per SupportingAccessPointName, inserisci il nome del tuo access point di supporto.

> **Note**
>
> Questo è un punto di accesso associato al bucket Amazon S3 scelto nel passaggio precedente. Se non disponi di punti di accesso associati al tuo bucket Amazon S3, puoi configurare il modello in modo che ne crei uno per te scegliendo true for. CreateNewSupportingAccessPoint

9. Seleziona Successivo.
10. Nella pagina Configure stack options (Configura opzioni pila), scegliere Next (Successivo).

Per ulteriori informazioni sulle impostazioni facoltative in questa pagina, consulta la sezione Impostazione delle opzioni dello stack AWS CloudFormation nella Guida per l'utente di AWS CloudFormation .

11. Nella pagina Revisione, scegliere Crea pila.

Usando il AWS Cloud Development Kit (AWS CDK)

Per ulteriori informazioni sulla configurazione degli access point Object Lambda utilizzando AWS CDK, *AWS::S3ObjectLambda*vedete Construct Library nell'AWS Cloud Development Kit (AWS CDK) API Reference.

Automatizza la configurazione di S3 Object Lambda con un modello CloudFormation

Puoi utilizzare un AWS CloudFormation modello per creare rapidamente un punto di accesso Amazon S3 Object Lambda. Il CloudFormation modello crea automaticamente le risorse pertinenti, configura i ruoli AWS Identity and Access Management (IAM) e imposta una AWS Lambda funzione che gestisce automaticamente le richieste tramite l'access point Object Lambda. Con il CloudFormation modello, puoi implementare le migliori pratiche, migliorare il tuo livello di sicurezza e ridurre gli errori causati dai processi manuali.

Questo GitHub repository contiene il CloudFormation modello e il codice sorgente della funzione Lambda. Per istruzioni su come utilizzare il modello, consulta the section called “Creazione di punti di accesso Object Lambda”.

La funzione Lambda fornita nel modello non esegue alcuna trasformazione. Al contrario, restituisce gli oggetti così come sono dal bucket S3. È possibile clonare la funzione e aggiungere il proprio codice di trasformazione per modificare ed elaborare i dati man mano che vengono restituiti ad un'applicazione. Per ulteriori informazioni sulla modifica della funzione, consulta the section called “Modifica della funzione Lambda” e the section called “Scrittura delle funzioni Lambda”.

Modificare il modello

Creazione di un nuovo punto di accesso di supporto

Lambda per oggetti S3 utilizza due punti di accesso, un punto di accesso Lambda per oggetti e un punto di accesso S3 standard, denominato punto di accesso di supporto. Quando effettui una richiesta a un punto di accesso Lambda per oggetti, S3 richiama Lambda per tuo conto o delega la richiesta al punto di accesso di supporto, a seconda della configurazione di Lambda per oggetti S3. È

possibile creare un nuovo punto di accesso di supporto passando il seguente parametro come parte del comando `aws cloudformation deploy` durante l'implementazione del modello.

```
CreateNewSupportingAccessPoint=true
```

Configurazione di un payload di funzione

È possibile configurare un payload per fornire dati supplementari alla funzione Lambda passando il seguente parametro come parte del comando `aws cloudformation deploy` al momento dell'implementazione del modello.

```
LambdaFunctionPayload="format=json"
```

Abilitare il CloudWatch monitoraggio di Amazon

Puoi abilitare il CloudWatch monitoraggio passando il seguente parametro come parte del `aws cloudformation deploy` comando durante la distribuzione del modello.

```
EnableCloudWatchMonitoring=true
```

Questo parametro abilita i parametri delle richieste Object Lambda Access Point per Amazon S3 e crea CloudWatch due allarmi per monitorare gli errori lato client e lato server.

> **Note**
>
> CloudWatch L'utilizzo di Amazon comporterà costi aggiuntivi. Per ulteriori informazioni sulle metriche delle richieste Amazon S3, consulta la sezione Monitoraggio e registrazione degli access point.
> Per i dettagli sui prezzi, vedere Prezzi di CloudWatch .

Configurazione della simultaneità fornita

Per ridurre la latenza, è possibile configurare la simultaneità con provisioning per la funzione Lambda che supporta il punto di accesso Lambda per oggetti modificando il modello per includere le seguenti righe in `Resources`.

```
LambdaFunctionVersion:
      Type: AWS::Lambda::Version
```

```
      Properties:
        FunctionName: !Ref LambdaFunction
        ProvisionedConcurrencyConfig:
            ProvisionedConcurrentExecutions: Integer
```

> **Note**
>
> Saranno applicati costi aggiuntivi la simultaneità con provisioning. Per ulteriori informazioni sulla simultaneità con provisioning, consulta Gestione della simultaneità con provisioning di Lambda nella Guida per gli sviluppatori di AWS Lambda .
> Per i dettagli sui prezzi, vedere Prezzi di AWS Lambda.

Modifica della funzione Lambda

Modifica dei valori di intestazione per una richiesta **`GetObject`**

Per impostazione predefinita, la funzione Lambda inoltra tutte le intestazioni, eccetto `Content-Length` ed `ETag`, dalla richiesta URL prefirmata al client `GetObject`. In base al codice di trasformazione nella funzione Lambda, puoi scegliere di inviare nuovi valori di intestazione al client `GetObject`.

È possibile aggiornare la funzione Lambda per inviare nuovi valori di intestazione passandoli nell'operazione API `WriteGetObjectResponse`.

Ad esempio, se la funzione Lambda traduce il testo negli oggetti Amazon S3 in una lingua diversa, puoi passare un nuovo valore nell'intestazione `Content-Language`. Puoi fare ciò modificando la funzione `writeResponse` come descritto di seguito.

```
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();
}
```

Per un elenco completo delle intestazioni supportate, consulta `WriteGetObjectResponse` nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Restituzione di intestazioni di metadati

È possibile aggiornare la funzione Lambda per inviare nuovi valori di intestazione passandoli nella richiesta dell'operazione API `WriteGetObjectResponse`.

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,
 transformedObject: Buffer,
  headers: Headers): Promise<PromiseResult<{}, AWSError>> {
  const { algorithm, digest } = getChecksum(transformedObject);

  return s3Client.writeGetObjectResponse({
    RequestRoute: requestContext.outputRoute,
    RequestToken: requestContext.outputToken,
    Body: transformedObject,
    Metadata: {
      'body-checksum-algorithm': algorithm,
      'body-checksum-digest': digest,
      'my-new-header': 'my-new-value'
    },
    ...headers
  }).promise();
}
```

Restituzione di un nuovo codice di stato

È possibile restituire un codice di stato personalizzato al client `GetObject` passandolo nella richiesta dell'operazione API `WriteGetObjectResponse`.

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,
 transformedObject: Buffer,
  headers: Headers): Promise<PromiseResult<{}, AWSError>> {
  const { algorithm, digest } = getChecksum(transformedObject);
```

```
  return s3Client.writeGetObjectResponse({
    RequestRoute: requestContext.outputRoute,
    RequestToken: requestContext.outputToken,
    Body: transformedObject,
    Metadata: {
      'body-checksum-algorithm': algorithm,
      'body-checksum-digest': digest
    },
    ...headers,
    StatusCode: Integer
  }).promise();
}
```

Per un elenco completo degli stati supportati, consulta `WriteGetObjectResponse` nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Applicazione dei **`Range`** e **`partNumber`** all'oggetto di origine

Per impostazione predefinita, l'Object Lambda Access Point creato dal CloudFormation modello può gestire i parametri `Range` and`partNumber`. La funzione Lambda applica l'intervallo o il numero di parte richiesto all'oggetto trasformato. A tale scopo, è necessario scaricare l'intero oggetto ed eseguire la trasformazione. In alcuni casi, gli intervalli di oggetti trasformati potrebbero essere associati esattamente agli intervalli di oggetti fonte. Ciò significa che la richiesta dell'intervallo di byte A-B sull'oggetto di origine e l'esecuzione della trasformazione potrebbero produrre lo stesso risultato della richiesta dell'intero oggetto, dell'esecuzione della trasformazione e della restituzione dell'intervallo di byte A-B sull'oggetto trasformato.

In questi casi, è possibile modificare l'implementazione della funzione Lambda per applicare l'intervallo o il numero di parte direttamente all'oggetto fonte. Questo approccio riduce la latenza generale della funzione e la memoria richieste. Per ulteriori informazioni, consulta the section called “Utilizzo delle intestazioni Range e partNumber”.

Disabilitazione della gestione di **`Range`** e**`partNumber`**

Per impostazione predefinita, l'Object Lambda Access Point creato dal CloudFormation modello può gestire i parametri `Range` and`partNumber`. Se questo comportamento non è necessario, è possibile disabilitarlo rimuovendo le seguenti righe dal modello:

```
AllowedFeatures:
  - GetObject-Range
  - GetObject-PartNumber
```

```
  - HeadObject-Range
  - HeadObject-PartNumber
```

Trasformazione di oggetti di grandi dimensioni

Per impostazione predefinita, la funzione Lambda elabora l'intero oggetto in memoria prima di poter avviare lo streaming della risposta a S3 Object Lambda. È possibile modificare la funzione per effettuare lo streaming della risposta mentre esegue la trasformazione. Ciò aiuta a ridurre la latenza della trasformazione e la dimensione della memoria della funzione Lambda. Per un'implementazione esemplificativa, consulta Streaming esemplificativo del contenuto compresso.

Utilizzo dei punti di accesso Amazon S3 Object Lambda

Le richieste tramite gli punti di accesso Lambda per oggetti Amazon S3 si effettuano esattamente come le richieste tramite altri punti di accesso. Per ulteriori informazioni su come effettuare le richieste tramite un punto di accesso, consulta Utilizzo degli access point. Puoi effettuare richieste tramite i punti di accesso Object Lambda utilizzando la console Amazon S3 AWS Command Line Interface ,AWS CLI() AWS , gli SDK o l'API REST di Amazon S3.

> **Important**
>
> Il nome della risorsa Amazon (ARN) del punto di accesso Lambda per oggetti utilizza un nome di servizio di `s3-object-lambda`. Quindi, gli ARN degli punti di accesso Lambda per oggetti iniziano con `arn:aws:s3-object-lambda`, anziché `arn:aws:s3`, che viene utilizzato per altri punti di accesso.

Come trovare l'ARN per un punto di accesso Lambda per oggetti

Per utilizzare un punto di accesso Object Lambda con AWS CLI o AWS SDK, devi conoscere l'Amazon Resource Name (ARN) del punto di accesso Object Lambda. Gli esempi seguenti mostrano come trovare l'ARN di un punto di accesso Lambda per oggetti utilizzando la console Amazon S3 o la AWS CLI.

Utilizzo della console S3

Per trovare l'ARN per un punto di accesso Lambda per oggetti

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

2. Nel pannello di navigazione a sinistra, scegli Object Lambda Access Points (Punti di accesso Object Lambda).
3. Seleziona il pulsante di opzione accanto al punto di accesso Lambda per oggetti di cui vuoi copiare l'ARN.
4. Scegli Copy ARN (Copia ARN).

Usando il AWS CLI

Per trovare l'ARN per il tuo punto di accesso Object Lambda utilizzando il AWS CLI

1. Per recuperare un elenco degli punti di accesso Lambda per oggetti associati al tuo Account AWS, esegui il comando riportato di seguito. Prima di eseguire il comando, sostituisci l'ID dell'account *111122223333*con il tuo Account AWS ID.

```
aws s3control list-access-points-for-object-lambda --account-id 111122223333
```

2. Esamina l'output del comando per trovare l'ARN del punto di accesso Lambda per oggetti che desideri utilizzare. L'output del comando precedente dovrebbe essere simile all'esempio seguente.

```
{
    "ObjectLambdaAccessPointList": [
        {
            "Name": "my-object-lambda-ap",
            "ObjectLambdaAccessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/my-object-lambda-ap"
        },
        ...
    ]
}
```

Come utilizzare un alias in stile bucket per il punto di accesso Lambda per oggetti del bucket S3

Quando crei un punto di accesso Lambda per oggetti, Amazon S3 genera automaticamente un alias univoco per il tuo punto di accesso Lambda per oggetti. Puoi utilizzare questo alias del punto di accesso al posto di un nome del bucket Amazon S3 o del nome della risorsa Amazon (ARN) del punto di accesso Lambda per oggetti in una richiesta per qualsiasi operazione del piano dati del

punto di accesso. Per un elenco di queste operazioni, consulta Compatibilità dei punti di accesso con i AWS servizi.

Un nome alias del punto di accesso Lambda per oggetti viene creato nello stesso spazio dei nomi di un bucket Amazon S3. Questo nome alias viene generato automaticamente e non può essere modificato. Per un punto di accesso Lambda per oggetti esistente, l'alias viene assegnato automaticamente. Un nome alias del punto di accesso Lambda per oggetti soddisfa tutti i requisiti di un nome bucket Amazon S3 valido e comprende le seguenti parti:

Object Lambda Access Point name prefix-*metadata*--ol-s3

> **Note**
>
> Il suffisso `--ol-s3` è riservato ai nomi alias dei punti di accesso Lambda per oggetti e non può essere utilizzato per i nomi dei bucket o dei punti di accesso Lambda per oggetti. Per ulteriori informazioni sulle regole di denominazione dei bucket Amazon S3, consulta Regole di denominazione dei bucket.

Negli esempi seguenti viene illustrato l'ARN e l'alias per un punto di accesso Lambda per oggetti denominato *my-object-lambda-access-point*.

- ARN: `arn:aws:s3-object-lambda:`*region*`:`*account-id*`:accesspoint/`*my-object-lambda-access-point*
- Alias del punto di accesso Lambda per oggetti: *my-object-lambda-acc-1a4n8yjrb3kda96f67zwrwiiuse1a*`--ol-s3`

Quando si utilizza un punto di accesso Lambda per oggetti, è possibile utilizzare il nome alias del punto di accesso Lambda per oggetti senza la necessità di modifiche estese al codice.

Quando si elimina un punto di accesso Lambda per oggetti, il nome alias del punto di accesso Lambda per oggetti diventa inattivo e non viene allocato.

Come trovare l'alias per il punto di accesso Lambda per oggetti

Utilizzo della console S3

Per trovare l'alias per il tuo punto di accesso Lambda per oggetti utilizzando la console

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Object Lambda Access Points (Punti di accesso Object Lambda).
3. Per il punto di accesso Lambda per oggetti che desideri utilizzare, copia il valore dell'alias del punto di accesso Lambda per oggetti.

Usando il AWS CLI

Quando crei un punto di accesso Lambda per oggetti, Amazon S3 genera automaticamente un nome alias del punto di accesso Lambda per oggetti, come mostrato nell'esempio seguente. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni. Per informazioni su come creare un punto di accesso Object Lambda utilizzando il AWS CLI, vedere. Per creare un Object Lambda Access Point utilizzando 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"
    }
}
```

Il nome alias del punto di accesso Lambda per oggetti generato ha due campi:

- Il campo `Value` è il valore dell'alias del punto di accesso Lambda per oggetti.
- Il campo `Status` è lo stato dell'alias del punto di accesso Lambda per oggetti. Se lo stato è `PROVISIONING`, Amazon S3 alloca l'alias del punto di accesso Lambda per oggetti, ma l'alias non è ancora pronto per l'uso. Se lo stato è `READY`, l'alias del punto di accesso Lambda per oggetti è stato allocato correttamente ed è pronto per l'uso.

Per ulteriori informazioni sul tipo di dati `ObjectLambdaAccessPointAlias` nella REST API, consulta CreateAccessPointForObjectLambda e ObjectLambdaAccessPointAlias nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Come utilizzare l'alias del punto di accesso Lambda per oggetti

Puoi utilizzare l'alias del punto di accesso Lambda per oggetti al posto di un nome bucket Amazon S3 per le operazioni elencate in Compatibilità dei punti di accesso con i AWS servizi.

L' AWS CLI esempio seguente del `get-bucket-location` comando utilizza l'alias del punto di accesso del bucket per restituire il valore in Regione AWS cui si trova il bucket. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
aws s3api get-bucket-location --bucket my-object-lambda-
acc-w7i37nq6xuzgax3jw3oqtifiusw2a--ol-s3

{
    "LocationConstraint": "us-west-2"
}
```

Se l'alias del punto di accesso Lambda per oggetti in una richiesta non è valido, viene restituito il codice di errore `InvalidAccessPointAliasError`. Per ulteriori informazioni su `InvalidAccessPointAliasError` consulta Elenco dei codici di errore nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Le limitazioni di un alias del punto di accesso Lambda per oggetti sono le stesse di un alias del punto di accesso. Per ulteriori informazioni sulle limitazioni di un alias del punto di accesso, consulta Limitazioni.

Considerazioni sulla sicurezza per i punti di accesso S3 Object Lambda

Con Amazon S3 Object Lambda, puoi eseguire trasformazioni personalizzate sui dati non appena escono da Amazon S3 utilizzando la scalabilità e la flessibilità di una piattaforma di elaborazione. AWS Lambda S3 e Lambda rimangono protetti per impostazione predefinita, ma per conservare questo livello di sicurezza è necessaria un'attenzione speciale da parte dell'autore della funzione Lambda. S3 Object Lambda richiede che tutti gli accessi siano effettuati da entità autenticate (nessun accesso anonimo) e su HTTPS.

Per ridurre i rischi per la sicurezza, è consigliabile:

- Definire l'ambito del ruolo di esecuzione della funzione Lambda in base a un set di autorizzazioni il più limitato possibile.
- Se possibile, assicurati che la funzione Lambda acceda ad Amazon S3 tramite l'URL prefirmato fornito.

Configurazione delle policy IAM

Gli access point S3 supportano policy relative alle risorse AWS Identity and Access Management (IAM) che consentono di controllare l'uso del punto di accesso in base alla risorsa, all'utente o ad altre condizioni. Per ulteriori informazioni, consulta Configurazione delle policy IAM per i punti di accesso Lambda per oggetti.

Funzionamento della crittografia

Poiché gli access point Object Lambda utilizzano sia Amazon S3 che Amazon S3 AWS Lambda, esistono differenze nel comportamento di crittografia. Per ulteriori informazioni sul comportamento della crittografia predefinita di S3, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3.

- Quando si utilizza la crittografia lato server S3 con i punti di accesso Lambda per oggetti, l'oggetto viene decrittato prima di essere inviato a Lambda. Dopo l'invio a Lambda, l'oggetto viene elaborato in modo non crittografato (nel caso di una richiesta `GET` o `HEAD`).
- Per evitare la registrazione della chiave di crittografia, S3 rifiuterà le richieste `GET` e `HEAD` relative agli oggetti crittografati utilizzando la crittografia lato server con chiavi fornite dal cliente (SSE-C). Tuttavia, la funzione Lambda può ancora recuperare questi oggetti a condizione che abbia accesso alla chiave fornita dal client.
- Quando utilizzi la crittografia lato client S3 con i punti di accesso Lambda per oggetti, assicurati che Lambda abbia accesso alla chiave di crittografia affinché possa decrittare ed eseguire nuovamente la crittografia dell'oggetto.

Sicurezza dei punti di accesso

Lambda per oggetti S3 utilizza due punti di accesso, un punto di accesso Lambda per oggetti e un punto di accesso S3 standard, denominato punto di accesso di supporto. Quando effettui una richiesta a un punto di accesso Lambda per oggetti, S3 richiama Lambda per tuo conto o delega la richiesta al punto di accesso di supporto, a seconda della configurazione di Lambda per oggetti S3. Quando Lambda viene richiamato per una richiesta, S3 genera un URL prefirmato per l'oggetto

per tuo conto tramite il punto di accesso di supporto. Quando viene richiamata, la funzione Lambda riceverà questo URL come input.

È possibile impostare la funzione Lambda in modo che utilizzi questo URL prefirmato per recuperare l'oggetto originale invece di richiamare direttamente S3. Questo modello consente di applicare limiti di sicurezza migliori agli oggetti. È possibile limitare l'accesso diretto agli oggetti tramite bucket S3 o punti di accesso S3 a un set limitato di ruoli o utenti IAM. Questo approccio protegge anche le funzioni Lambda dall'essere soggette al problema del "confused deputy", in cui una funzione configurata erroneamente con autorizzazioni diverse rispetto all'invoker potrebbe consentire o negare l'accesso agli oggetti quando non dovrebbe.

Accesso pubblico ai punti di accesso Object Lambda

S3 Object Lambda non consente l'accesso anonimo o pubblico perché Amazon S3 deve autorizzare l'identità per completare qualsiasi richiesta di S3 Object Lambda. Quando si richiamano le richieste tramite un punto di accesso Lambda per oggetti, è necessaria l'autorizzazione `lambda:InvokeFunction` per la funzione Lambda configurata. Allo stesso modo, quando si richiamano altre operazioni di API tramite un punto di accesso Lambda per oggetti, è necessario disporre delle autorizzazioni `s3:*`.

Senza queste autorizzazioni, le richieste per richiamare Lambda o delegare a S3 avranno esito negativo e verrà restituito un errore HTTP 403 Accesso negato. Tutti gli accessi devono essere effettuati da principali autenticati. Se hai bisogno di un accesso pubblico, come possibile alternativa può essere utilizzato Lambda@Edge. Per ulteriori informazioni, consulta Customizing at the edge with Lambda @Edge nella CloudFront Amazon Developer Guide.

Indirizzi IP dei punti di accesso Lambda per oggetti

Le sottoreti `describe-managed-prefix-lists` supportano gli endpoint VPC (Virtual Private Cloud) del gateway e sono correlate alla tabella di instradamento degli endpoint VPC. Poiché il punto di accesso per le espressioni Lambda dell'oggetto non supporta il VPC del gateway, i relativi intervalli IP mancano. Gli intervalli mancanti appartengono ad Amazon S3, ma non sono supportati dagli endpoint VPC del gateway. Per ulteriori informazioni DescribeManagedPrefixListsin merito`describe-managed-prefix-lists`, consulta il riferimento alle API di Amazon EC2 e gli intervalli di indirizzi AWS IP nel. Riferimenti generali di AWS

Configurazione delle policy IAM per i punti di accesso Lambda per oggetti

I punti di accesso Amazon S3 supportano le policy delle risorse AWS Identity and Access Management (IAM) che puoi utilizzare per controllare l'uso del punto di accesso in base alla risorsa,

all'utente o ad altre condizioni. È possibile controllare l'accesso tramite una policy di risorse opzionale sul punto di accesso Lambda per oggetti o una policy di risorse sul punto di accesso di supporto. Per step-by-step esempi, consulta Tutorial: trasformazione dei dati per l'applicazione con S3 Object Lambda eTutorial: rilevamento e oscuramento dei dati PII con S3 Object Lambda e Amazon Comprehend.

Per utilizzare i punti di accesso Lambda per oggetti, le seguenti quattro risorse devono disporre delle seguenti autorizzazioni:

- L'identità IAM, ad esempio un utente o un ruolo. Per ulteriori informazioni sulle identità IAM e sulle best practice, consulta Identità IAM (utenti, gruppi di utenti e ruoli) nella Guida per l'utente di IAM.
- Il bucket e il relativo punto di accesso standard associato Quando utilizzi i punti di accesso Lambda per oggetti, questo punto di accesso standard è noto come punto di accesso di supporto.
- Il punto di accesso Lambda per oggetti.
- La AWS Lambda funzione.

> **Important**
>
> Prima di salvare la politica, assicurati di risolvere gli avvisi di sicurezza, gli errori, gli avvisi generali e i suggerimenti di. AWS Identity and Access Management Access Analyzer IAM Access Analyzer esegue controlli della policy per convalidarla in rapporto alla sintassi della policy e alle best practice di IAM. Questi controlli generano risultati e forniscono suggerimenti utili per aiutarti a creare policy funzionali e conformi alle best practice per la sicurezza. Per ulteriori informazioni sulla convalida delle policy tramite IAM Access Analyzer, consulta Convalida delle policy di IAM Access Analyzer nella Guida per l'utente di IAM. Per visualizzare un elenco di avvisi, errori e suggerimenti di IAM Access Analyzer, consulta Riferimento ai controlli delle policy IAM Access Analyzer.

In questi esempi di policy si presuppone di disporre delle seguenti risorse:

- Un bucket Amazon S3 con il seguente nome della risorsa Amazon (ARN):

  `arn:aws:s3:::DOC-EXAMPLE-BUCKET1`
- Un punto di accesso standard Amazon S3 su questo bucket con il seguente ARN:

  `arn:aws:s3:us-east-1:111122223333:accesspoint/my-access-point`

- Un punto di accesso Lambda per oggetti con il seguente ARN:

  `arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/my-object-lambda-ap`

- Una AWS Lambda funzione con il seguente ARN:

  `arn:aws:lambda:us-east-1:111122223333:function:MyObjectLambdaFunction`

> **Note**
>
> Se utilizzi una funzione Lambda dal tuo account, devi includere la versione della funzione specifica nella tua dichiarazione politica. *Nel seguente esempio ARN, la versione è indicata da 1:*
> `arn:aws:lambda:us-east-1:111122223333:function:MyObjectLambdaFunction:1`
> Lambda non supporta l'aggiunta di policy IAM alla versione. $LATEST Per ulteriori informazioni sulle versioni delle funzioni Lambda, consulta Versioni delle funzioni Lambda nella Guida per gli sviluppatori di AWS Lambda .

Example : policy di bucket che delega il controllo degli accessi ai punti di accesso standard

Il seguente esempio di policy di bucket S3 delega il controllo degli accessi di un bucket ai relativi punti di accesso standard. Questa policy consente l'accesso completo a tutti i punti di accesso di proprietà dell'account del proprietario del bucket. Pertanto, tutto l'accesso a questo bucket è controllato dalle policy associate ai punti di accesso. Gli utenti possono eseguire la lettura dal bucket solo mediante un punto di accesso; ciò significa che le operazioni possono essere richiamate solo tramite i punti di accesso. Per ulteriori informazioni, consulta Delegazione del controllo di accesso agli access point.

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS":"account-ARN"},
        "Action" : "*",
        "Resource" : [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET1",
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*"
```

```
        ],
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointAccount" : "Bucket owner's account
 ID" }
        }
    }]
}
```

Example — Policy IAM che concede a un utente le autorizzazioni necessarie per utilizzare un punto di accesso Object Lambda

La seguente policy IAM concede a un utente le autorizzazioni per la funzione Lambda, il punto di accesso standard e il punto di accesso Lambda per oggetti.

```
{
  "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"
    }
  ]
}
```

Abilitazione delle autorizzazioni per i ruoli di esecuzione Lambda

Quando vengono effettuate `GET` richieste a un punto di accesso Object Lambda, la funzione Lambda necessita dell'autorizzazione per inviare dati a S3 Object Lambda Access Point. Per fornire questa autorizzazione, abilita l'autorizzazione `s3-object-lambda:WriteGetObjectResponse` sul ruolo di esecuzione della funzione Lambda. Puoi creare un nuovo ruolo di esecuzione o aggiornare un ruolo esistente.

> **Note**
>
> La funzione richiede l'autorizzazione `s3-object-lambda:WriteGetObjectResponse` solo se stai effettuando una richiesta `GET`.

Per creare un ruolo di esecuzione nella console IAM

1. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra, seleziona Ruoli.
3. Scegli Crea ruolo.
4. In Common use cases (Casi di utilizzo comuni), scegliere Lambda.
5. Seleziona Successivo.

6. Nella pagina Aggiungi autorizzazioni, cerca la policy AWS gestita `AmazonS3ObjectLambdaExecutionRolePolicy`, quindi seleziona la casella di controllo accanto al nome della policy.

   Questa politica dovrebbe contenere l'operazione `s3-object-lambda:WriteGetObjectResponse`.
7. Seleziona Successivo.
8. Nella pagina Name, review, and create (Denomina, rivedi e crea), in Role name (Nome ruolo) immetti **`s3-object-lambda-role`**.
9. (Facoltativo) Aggiungi una descrizione e i tag per questo ruolo.
10. Scegli Crea ruolo.
11. Applica il nuovo **`s3-object-lambda-role`** quale ruolo di esecuzione della funzione Lambda. Questa operazione può essere eseguita durante o dopo la creazione della funzione Lambda nella console Lambda.

Per ulteriori informazioni sui ruoli di esecuzione, consulta la sezione Ruolo di esecuzione Lambda nella Guida per gli sviluppatori di AWS Lambda .

Utilizzo delle chiavi di contesto con i punti di accesso Lambda per oggetti

S3 Object Lambda valuterà le chiavi di contesto come `s3-object-lambda:TlsVersion` o `s3-object-lambda:AuthType` in base alla connessione o alla firma della richiesta. Tutte le altre chiavi di contesto, ad esempio `s3:prefix`, vengono valutate da Amazon S3.

Supporto CORS per Punto di accesso per le espressioni Lambda dell'oggetto

Quando Lambda per oggetti Amazon S3 riceve una richiesta da un browser o la richiesta include un'intestazione `Origin`, Lambda per oggetti Amazon S3 aggiunge sempre un campo di intestazione `“AllowedOrigins":"*"`.

Per ulteriori informazioni, consulta Utilizzo della funzionalità Cross-Origin Resource Sharing (CORS).

Scrittura di funzioni Lambda per i punti di accesso Lambda per oggetti S3

Questa sezione descrive in dettaglio come scrivere AWS Lambda funzioni da utilizzare con gli access point Amazon S3 Object Lambda.

Per informazioni sulle end-to-end procedure complete per alcune attività di S3 Object Lambda, consulta quanto segue:

- Tutorial: trasformazione dei dati per l'applicazione con S3 Object Lambda
- Tutorial: rilevamento e oscuramento dei dati PII con S3 Object Lambda e Amazon Comprehend
- Tutorial: utilizzo di S3 Object Lambda per aggiungere filigrane alle immagini in modo dinamico man mano che vengono recuperate

Argomenti

- Utilizzo di richieste GetObject in Lambda
- Utilizzo di richieste HeadObject in Lambda
- Utilizzo di richieste ListObjects in Lambda
- Utilizzo di richieste ListObjectsV2 in Lambda
- Formato e utilizzo del contesto degli eventi
- Utilizzo delle intestazioni Range e partNumber

Utilizzo di richieste **`GetObject`** in Lambda

Questa sezione presuppone che il punto di accesso Lambda per oggetti sia configurato per richiamare la funzione Lambda per `GetObject`. S3 Object Lambda include l'operazione API Amazon S3 `WriteGetObjectResponse`, che consente alla funzione Lambda di fornire dati personalizzati e intestazioni di risposta al chiamante `GetObject`.

`WriteGetObjectResponse` offre un ampio controllo su codice di stato, intestazioni di risposta e corpo della risposta, in base ai requisiti di elaborazione. È possibile utilizzare `WriteGetObjectResponse` per rispondere con l'intero oggetto trasformato, con parti dell'oggetto trasformato o con altre risposte in base al contesto dell'applicazione. Nella sezione seguente sono illustrati esempi univoci di utilizzo dell'operazione API `WriteGetObjectResponse`.

- Esempio 1: risposta con un codice di stato HTTP 403 (Forbidden) (Accesso negato)
- Esempio 2: Rispondere con un'immagine trasformata
- Esempio 3: Streaming di contenuto compresso

Esempio 1: risposta con un codice di stato HTTP 403 (Forbidden) (Accesso negato)

È possibile utilizzare `WriteGetObjectResponse` per rispondere con il codice di stato HTTP 403 (Non consentito) in base al contenuto dell'oggetto.

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 };
}
```

Esempio 2: Rispondere con un'immagine trasformata

Durante la trasformazione dell'immagine, è possibile che siano necessari tutti i byte dell'oggetto di fonte prima di poter iniziare a elaborarli. In questo caso, la tua richiesta `WriteGetObjectResponse` restituisce l'intero oggetto all'applicazione richiedente in una sola chiamata.

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 };
}
```

Esempio 3: Streaming di contenuto compresso

Durante la compressione degli oggetti, i dati compressi vengono prodotti in modo incrementale. Di conseguenza, puoi utilizzare la richiesta `WriteGetObjectResponse` per restituire i dati compressi non appena sono pronti. Come mostrato in questo esempio, non è necessario conoscere la lunghezza della trasformazione completata.

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

Sebbene S3 Object Lambda consente fino a 60 secondi per inviare una risposta completa al chiamante tramite la richiesta `WriteGetObjectResponse`, la quantità effettiva di tempo disponibile potrebbe essere inferiore. Ad esempio, il timeout della funzione Lambda potrebbe essere inferiore a 60 secondi. In altri casi, il chiamante potrebbe avere timeout più rigorosi.

Affinché il chiamante originale riceva una risposta diversa dal codice di stato HTTP 500 (Internal Server Error) (Errore interno del server), la chiamata `WriteGetObjectResponse` deve essere completata. Se la funzione Lambda restituisce un risultato, eccezionalmente o in altro modo, prima che l'operazione API `WriteGetObjectResponse` venga richiamata, il chiamante originale riceverà una risposta 500 (Internal Server Error) (Errore interno del server). Le eccezioni generate durante il tempo necessario per completare la risposta comportano risposte troncate al chiamante. Se la funzione Lambda riceve una risposta con codice di stato HTTP 200 (OK) dalla chiamata API `WriteGetObjectResponse`, il chiamante originale ha inviato la richiesta completa. La risposta della funzione Lambda, indipendentemente dal fatto che un'eccezione sia generata o meno, viene ignorata da S3 Object Lambda.

Quando viene richiamata l'operazione API `WriteGetObjectResponse`, Amazon S3 richiede il token dell'instradamento e della richiesta dal contesto dell'evento. Per ulteriori informazioni, consulta Formato e utilizzo del contesto degli eventi.

I parametri relativi ai token dell'instradamento e della richiesta sono necessari per collegare la risposta `WriteGetObjectResult` al chiamante originale. Sebbene sia sempre opportuno riprovare le risposte 500 (Internal Server Error) (Errore interno del server), è necessario considerare che il token della richiesta è un token monouso e i successivi tentativi di utilizzo possono comportare risposte con codice di stato 400 (Bad Request) (Richiesta non valida). Anche se la chiamata a `WriteGetObjectResponse` con i token dell'instradamento e della richiesta non ha bisogno di essere effettuata dalla funzione Lambda richiamata, deve essere effettuata da un'identità nello stesso account. La chiamata deve anche essere completata prima che la funzione Lambda finisca l'esecuzione.

Utilizzo di richieste **`HeadObject`** in Lambda

Questa sezione presuppone che il punto di accesso Lambda per oggetti sia configurato per richiamare la funzione Lambda per `HeadObject`. Lambda riceverà un payload JSON contenente una chiave chiamata `headObjectContext`. All'interno del contesto, esiste un'unica proprietà chiamata `inputS3Url`, che è un URL prefirmato per il punto di accesso di supporto per `HeadObject`.

L'URL prefirmato includerà le seguenti proprietà, se specificate:

- `versionId` (nei parametri della query)
- `requestPayer` (nell'intestazione `x-amz-request-payer`)
- `expectedBucketOwner` (nell'intestazione `x-amz-expected-bucket-owner`)

Le altre proprietà non saranno prefirmate e quindi non saranno incluse. Le opzioni non firmate inviate come intestazioni possono essere aggiunte manualmente alla richiesta quando si richiama l'URL prefirmato che si trova nelle intestazioni `userRequest`. Le opzioni di crittografia lato server non sono supportate per `HeadObject`.

Per i parametri URI della sintassi della richiesta, consulta `HeadObject` nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Il seguente esempio mostra un payload di input Lambda JSON per `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 funzione Lambda dovrebbe restituire un oggetto JSON contenente le intestazioni e i valori che verranno restituiti per la chiamata `HeadObject`.

Il seguente esempio illustra la struttura dell'oggetto JSON della risposta Lambda per `HeadObject`.

```
{
    "statusCode": <number>; // Required
    "errorCode": <string>;
    "errorMessage": <string>;
    "headers": {
        "Accept-Ranges": <string>,
        "x-amz-archive-status": <string>,
        "x-amz-server-side-encryption-bucket-key-enabled": <boolean>,
        "Cache-Control": <string>,
        "Content-Disposition": <string>,
        "Content-Encoding": <string>,
        "Content-Language": <string>,
        "Content-Length": <number>, // Required
        "Content-Type": <string>,
        "x-amz-delete-marker": <boolean>,
        "ETag": <string>,
        "Expires": <string>,
        "x-amz-expiration": <string>,
        "Last-Modified": <string>,
        "x-amz-missing-meta": <number>,
        "x-amz-object-lock-mode": <string>,
        "x-amz-object-lock-legal-hold": <string>,
        "x-amz-object-lock-retain-until-date": <string>,
        "x-amz-mp-parts-count": <number>,
        "x-amz-replication-status": <string>,
```

```
        "x-amz-request-charged": <string>,
        "x-amz-restore": <string>,
        "x-amz-server-side-encryption": <string>,
        "x-amz-server-side-encryption-customer-algorithm": <string>,
        "x-amz-server-side-encryption-aws-kms-key-id": <string>,
        "x-amz-server-side-encryption-customer-key-MD5": <string>,
        "x-amz-storage-class": <string>,
        "x-amz-tagging-count": <number>,
        "x-amz-version-id": <string>,
        <x-amz-meta-headers>: <string>, // user-defined metadata
        "x-amz-meta-meta1": <string>, // example of the user-defined metadata header,
 it will need the x-amz-meta prefix
        "x-amz-meta-meta2": <string>
        ...
    };
}
```

L'esempio seguente mostra come utilizzare l'URL prefirmato per compilare la risposta modificando i valori dell'intestazione secondo necessità prima di restituire l'oggetto 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
        }
```

Utilizzo di richieste **`ListObjects`** in Lambda

Questa sezione presuppone che il punto di accesso Lambda per oggetti sia configurato per richiamare la funzione Lambda per `ListObjects`. Lambda riceverà il payload JSON con un nuovo oggetto denominato `listObjectsContext`. `listObjectsContext` contiene un'unica proprietà `inputS3Url`, che è un URL prefirmato per il punto di accesso di supporto per `ListObjects`.

A differenza di `GetObject` e `HeadObject`, l'URL prefirmato includerà le seguenti proprietà, se specificate:

- Tutti i parametri della query
- `requestPayer` (nell'intestazione `x-amz-request-payer`)
- `expectedBucketOwner` (nell'intestazione `x-amz-expected-bucket-owner`)

Per i parametri URI della sintassi della richiesta, consulta `ListObjects` nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

> **Important**
>
> Ti consigliamo di utilizzare la versione più recente, ListObjectsV2, per lo sviluppo di applicazioni. Per la compatibilità con le versioni precedenti, Amazon S3 continua a supportare `ListObjects`.

Il seguente esempio illustra il payload di input Lambda JSON per `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 funzione Lambda deve restituire un oggetto JSON contenente il codice di stato, il risultato XML dell'elenco o le informazioni sull'errore che verranno restituite da S3 Object Lambda.

S3 Object Lambda non elabora né convalida `listResultXml`, ma lo inoltra al chiamante `ListObjects`. Per `listBucketResult`, S3 Object Lambda si aspetta che determinate proprietà siano di un tipo specifico e genererà eccezioni se non è in grado di analizzarle. `listResultXml` `elistBucketResult` non possono essere specificati contemporaneamente.

L'esempio seguente illustra come utilizzare l'URL prefirmato per richiamare Amazon S3 e utilizzare il risultato per compilare una risposta, incluso il controllo degli errori.

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
```

Il seguente esempio illustra la struttura dell'oggetto JSON della risposta Lambda per `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'
        },
        ...
        ],
    }
}
```

Utilizzo di richieste **`ListObjectsV2`** in Lambda

Questa sezione presuppone che il punto di accesso Lambda per oggetti sia configurato per richiamare la funzione Lambda per `ListObjectsV2`. Lambda riceverà il payload JSON con un nuovo oggetto denominato `listObjectsV2Context`. `listObjectsV2Context` contiene un'unica proprietà `inputS3Url`, che è un URL prefirmato per il punto di accesso di supporto per `ListObjectsV2`.

A differenza di `GetObject` e `HeadObject`, l'URL prefirmato includerà le seguenti proprietà, se specificate:

- Tutti i parametri della query
- `requestPayer` (nell'intestazione `x-amz-request-payer`)
- `expectedBucketOwner` (nell'intestazione `x-amz-expected-bucket-owner`)

Per i parametri URI della sintassi della richiesta, consulta `ListObjectsV2` nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Il seguente esempio illustra il payload di input Lambda JSON per `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 funzione Lambda deve restituire un oggetto JSON contenente il codice di stato, il risultato XML dell'elenco o le informazioni sull'errore che verranno restituite da S3 Object Lambda.

S3 Object Lambda non elabora né convalida `listResultXml`, ma lo inoltra al chiamante `ListObjectsV2`. Per `listBucketResult`, S3 Object Lambda si aspetta che determinate proprietà siano di un tipo specifico e genererà eccezioni se non è in grado di analizzarle. `listResultXml` `elistBucketResult` non possono essere specificati contemporaneamente.

L'esempio seguente illustra come utilizzare l'URL prefirmato per richiamare Amazon S3 e utilizzare il risultato per compilare una risposta, incluso il controllo degli errori.

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
```

Il seguente esempio illustra la struttura dell'oggetto JSON della risposta Lambda per `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 e utilizzo del contesto degli eventi

Amazon S3 Object Lambda fornisce un contesto sulla richiesta che viene effettuata nel caso in cui venga passata alla tua funzione. AWS Lambda Il risultato è illustrato nello screenshot seguente. Le descrizioni dei campi sono riportate dopo l'esempio.

```
{
    "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"
}
```

I seguenti campi sono inclusi nella richiesta:

- `xAmzRequestId`: l'ID della richiesta di Amazon S3 per questa richiesta. Si consiglia di registrare questo valore per facilitare il debug.
- `getObjectContext`: i dettagli di input e output per le connessioni ad Amazon S3 e S3 Object Lambda.
  - `inputS3Url`: un URL prefirmato che può essere utilizzato per recuperare l'oggetto originale da Amazon S3. L'URL viene firmato utilizzando l'identità del chiamante originale e quando viene utilizzato l'URL vengono applicate le autorizzazioni dell'utente associato. Se nell'URL sono presenti intestazioni firmate, la funzione Lambda deve includerle nella chiamata ad Amazon S3, ad eccezione dell'intestazione `Host`.
  - `outputRoute` - Un token di routing che viene aggiunto all'URL di S3 Object Lambda quando la funzione Lambda richiama `WriteGetObjectResponse`.
  - `outputToken`: un token opaco utilizzato da S3 Object Lambda per abbinare la chiamata `WriteGetObjectResponse` al chiamante originale.
- `configuration`: informazioni di configurazione sul punto di accesso Lambda per oggetti.
  - `accessPointArn`: il nome della risorsa Amazon (ARN) del punto di accesso Lambda per oggetti che ha ricevuto questa richiesta.

  - `supportingAccessPointArn`: l'ARN del punto di accesso di supporto specificato nella configurazione del punto di accesso Lambda per oggetti.
  - `payload`: dati personalizzati applicati alla configurazione del punto di accesso Lambda per oggetti. S3 Object Lambda tratta questi dati come una stringa opaca, quindi potrebbe essere necessario decodificarli prima dell'utilizzo.
- `userRequest`: informazioni sulla chiamata originale a S3 Object Lambda.
  - `url`: l'URL decodificato della richiesta come ricevuto da S3 Object Lambda, esclusi eventuali parametri di query relativi all'autorizzazione.
  - `headers`: una mappa di stringa alle stringhe contenenti le intestazioni HTTP e i relativi valori dalla chiamata originale, escluse eventuali intestazioni relative all'autorizzazione. Se la stessa intestazione viene visualizzata più volte, i valori di ogni istanza della stessa intestazione vengono combinati in un elenco delimitato da virgole. Il formato maiuscolo/minuscolo delle intestazioni originali viene mantenuto in questa mappa.
- `userIdentity`: dettagli sull'identità che ha effettuato la chiamata a S3 Object Lambda. Per ulteriori informazioni, consulta Registrazione di eventi di dati per i percorsi nella Guida per l'utente di AWS CloudTrail .
  - `type`: il tipo di identità.
  - `accountId`— Account AWS A cui appartiene l'identità.
  - `userName`: il nome descrittivo dell'identità che ha effettuato la chiamata.
  - `principalId`: l'identificatore univoco per l'identità che ha effettuato la chiamata.
  - `arn`: l'ARN del principale che ha effettuato la chiamata. L'ultima sezione dell'ARN contiene l'utente o il ruolo che ha effettuato la chiamata.
  - `sessionContext`: se la richiesta è stata effettuata con le credenziali di sicurezza temporanee, questo elemento fornisce informazioni sulla sessione creata per tali credenziali.
  - `invokedBy`— Il nome di chi Servizio AWS ha effettuato la richiesta, ad esempio Amazon EC2 Auto Scaling o. AWS Elastic Beanstalk
  - `sessionIssuer`: se la richiesta è stata effettuata con le credenziali di sicurezza temporanee, questo elemento fornisce informazioni su come sono state ottenute tali credenziali.
- `protocolVersion`: l'ID versione del contesto fornito. Il formato di questo campo è `{Major Version}.{Minor Version}`. I numeri di versione secondari sono sempre numeri a due cifre. Qualsiasi rimozione o modifica alla semantica di un campo necessita un aumento della versione principale e richiede l'opt-in attivo. Amazon S3 può aggiungere nuovi campi in qualsiasi momento e in quel punto si potrebbe riscontrare un bump di versione minore. A causa della natura

delle implementazioni software, più versioni secondarie potrebbero essere visualizzati in uso contemporaneamente.

Utilizzo delle intestazioni Range e partNumber

Quando vengono utilizzati oggetti di grandi dimensioni in Amazon S3 Object Lambda, è possibile utilizzare l'intestazione HTTP `Range` per scaricare un intervallo di byte specificato da un oggetto. Puoi utilizzare connessioni simultanee ad Amazon S3 per recuperare diversi intervalli di byte all'interno dello stesso oggetto. Puoi inoltre specificare il parametro `partNumber` (un numero intero compreso tra 1 e 10.000) che esegue una richiesta basata su intervallo per la parte specificata dell'oggetto.

Perché ci sono diversi modi in cui potresti voler gestire una richiesta che include i parametri `Range` o `partNumber`, S3 Object Lambda non applica questi parametri all'oggetto trasformato. Al contrario, la AWS Lambda funzione deve implementare questa funzionalità in base alle esigenze dell'applicazione.

Per utilizzare i parametri `Range` e `partNumber` con S3 Object Lambda, procedi come segue:

- Abilita questi parametri nella configurazione del punto di accesso Lambda per oggetti.
- Scrivi una funzione Lambda in grado di gestire le richieste contenente questi parametri.

Di seguito viene descritto come realizzarlo.

Fase 1: configura il punto di accesso Lambda per oggetti

Per impostazione predefinita, gli punti di accesso Lambda per oggetti rispondono con un errore con codice di stato HTTP 501 (Not Implemented) a qualsiasi richiesta `GetObject` o `HeadObject` contenente un parametro `Range` o `partNumber` nelle intestazioni o nei parametri di query.

Per abilitare un punto di accesso Lambda per oggetti ad accettare tali richieste, devi includere `GetObject-Range`, `GetObject-PartNumber`, `HeadObject-Range` o `HeadObject-PartNumber` nella sezione `AllowedFeatures` della configurazione del punto di accesso Lambda per oggetti. Per ulteriori informazioni sull'aggiornamento della configurazione del punto di accesso Lambda per oggetti, consulta Creazione di punti di accesso Object Lambda.

Fase 2: implementa la gestione di **`Range`** o **`partNumber`** nella funzione Lambda

Quando il punto di accesso Lambda per oggetti richiama la funzione Lambda con una richiesta `GetObject` o `HeadObject` basata su intervallo, il parametro `Range` o `partNumber` è incluso nel contesto dell'evento. La posizione del parametro nel contesto dell'evento dipende dal parametro utilizzato e dal modo in cui è stato incluso nella richiesta originale al punto di accesso Lambda per oggetti, come illustrato nella tabella seguente.

Parametro	Posizione del contesto dell'evento
`Range` (intestazione)	`userRequest.headers.Range`
`Range` (parametro di query)	`userRequest.url` (`Range` del parametro di query)
`partNumber`	`userRequest.url` (`partNumber` del parametro di query)

> **Important**
>
> L'URL prefirmato fornito per il punto di accesso Lambda per oggetti non contiene il parametro `Range` o `partNumber` della richiesta originale. Vedi le seguenti opzioni su come gestire questi parametri nella tua AWS Lambda funzione.

Dopo aver estratto il valore `Range` o `partNumber`, è possibile adottare uno dei seguenti approcci in base alle esigenze dell'applicazione:

A. Mappare il valore **`Range`** o **`partNumber`** richiesto all'oggetto trasformato (consigliato).

Per gestire le richieste `Range` o `partNumber` nel modo più affidabile, procedi come segue:

- Recupera l'oggetto completo da Amazon S3.
- Trasforma l'oggetto.
- Applica i parametri `Range` o `partNumber` obbligatori all'oggetto trasformato.

Per fare ciò, utilizza l'URL prefirmato fornito per recuperare l'intero oggetto da Amazon S3 e quindi elaborare l'oggetto secondo necessità. Per un esempio di funzione Lambda che elabora un `Range` parametro in questo modo, guarda questo esempio nel repository AWS Samples GitHub .

B. Mappatura del **`Range`** richiesto all'URL prefirmato URL.

In alcuni casi, la funzione Lambda può mappare il valore o il `Range` richiesto direttamente all'URL prefirmato per recuperare solo parte dell'oggetto da Amazon S3. Questo approccio è appropriato solo se la trasformazione soddisfa entrambi i seguenti criteri:

1. La funzione di trasformazione può essere applicata a intervalli di oggetti parziali.
2. Applicando il parametro `Range` prima o dopo la funzione di trasformazione produce lo stesso oggetto trasformato.

Ad esempio, una funzione di trasformazione che converte tutti i caratteri di un oggetto con codifica ASCII in maiuscolo soddisfa entrambi i criteri precedenti. La trasformazione può essere applicata a una parte di un oggetto e applicando il parametro `Range` prima della trasformazione si ottiene lo stesso risultato dell'applicazione del parametro dopo la trasformazione.

Al contrario, una funzione che inverte i caratteri in un oggetto con codifica ASCII non soddisfa questi criteri. Tale funzione soddisfa il criterio 1, poiché può essere applicata a intervalli di oggetti parziali. Tuttavia, non soddisfa il criterio 2, perché l'applicazione del parametro `Range` prima che la trasformazione raggiunga risultati diversi rispetto all'applicazione del parametro dopo la trasformazione.

Considera una richiesta di applicare la funzione ai primi tre caratteri di un oggetto con il contenuto `abcdefg`. L'applicazione del parametro `Range` prima della trasformazione recupera solo `abc` e poi inverte i dati, restituendo `cba`. Ma se il parametro viene applicato dopo la trasformazione, la funzione recupera l'intero oggetto, lo inverte e quindi applica il parametro `Range`, restituendo `gfe`. Poiché questi risultati sono diversi, questa funzione non dovrebbe applicare il parametro `Range` durante il recupero dell'oggetto da Amazon S3. Invece, dovrebbe recuperare l'intero oggetto, eseguire la trasformazione e solo successivamente applicare il parametro `Range`.

> **Warning**
>
> In molti casi, l'applicazione del parametro `Range` o dell'URL prefirmato risulterà in un comportamento imprevisto da parte della funzione Lambda o del client richiedente. A meno che non sia sicuro che la tua applicazione funzioni correttamente quando recuperi solo un

oggetto parziale da Amazon S3, ti consigliamo di recuperare e trasformare oggetti completi come descritto in precedenza nell'approccio A.

Se l'applicazione soddisfa i criteri descritti in precedenza nell'approccio B, è possibile semplificare la AWS Lambda funzione recuperando solo l'intervallo di oggetti richiesto e quindi eseguendo la trasformazione su quell'intervallo.

Il seguente esempio di codice Java illustra come eseguire le seguenti operazioni:

- Recuperare l'intestazione `Range` dalla richiesta `GetObject`.
- Aggiungere l'intestazione `Range` all'URL prefirmato che Lambda può utilizzare per recuperare l'intervallo richiesto da 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;
}
```

Utilizzo delle AWS funzioni Lambda integrate

AWS fornisce alcune AWS Lambda funzioni predefinite che puoi utilizzare con Amazon S3 Object Lambda per rilevare e redigere informazioni di identificazione personale (PII) e decomprimere oggetti S3. Queste funzioni Lambda sono disponibili nel AWS Serverless Application Repository. È possibile selezionare queste funzioni mediante la AWS Management Console quando si crea il punto di accesso Lambda per oggetti.

Per ulteriori informazioni su come distribuire applicazioni serverless da, consulta Deploying Applications nella Developer Guide. AWS Serverless Application RepositoryAWS Serverless Application Repository

> **Note**
>
> I seguenti esempi possono essere utilizzati solo con richieste `GetObject`.

Esempio 1: Controllo degli accessi alle informazioni personali di identificazione (PII)

Questa funzione Lambda utilizza Amazon Comprehend, un servizio di elaborazione del linguaggio naturale (NLP) basato sul machine learning per trovare informazioni e relazioni in un testo. Questa funzionalità rileva automaticamente le informazioni personali di identificazione (PII) come nomi, indirizzi, date, numeri di carta di credito e numeri di previdenza sociale nei documenti presenti in un bucket Amazon S3. Se nel bucket sono presenti documenti che includono questo tipo di informazioni, è possibile configurare la funzione di controllo degli accessi alle queste informazioni di S3 Object Lambda per rilevare questi tipi di entità PII e bloccare l'accesso agli utenti non autorizzati.

Per iniziare, è sufficiente implementare la seguente funzione Lambda nell'account in uso e aggiungere il nome della risorsa Amazon (ARN) della funzione nella configurazione del punto di accesso Lambda per oggetti.

Di seguito è riportato un esempio di ARN per questa funzione:

```
arn:aws:serverlessrepo:us-east-1:111122223333:applications/
ComprehendPiiAccessControlS3ObjectLambda
```

È possibile aggiungere o visualizzare questa funzione su AWS Management Console utilizzando il seguente AWS Serverless Application Repository link: S3. ComprehendPiiAccessControl ObjectLambda

Per visualizzare questa funzione su GitHub, consulta Amazon Comprehend S3 Object Lambda.

Esempio 2: oscuramento di PII

Questa funzione Lambda utilizza Amazon Comprehend, un servizio di elaborazione del linguaggio naturale (NLP) basato sul machine learning per trovare informazioni e relazioni in un testo. Questa funzione oscura automaticamente le informazioni personali di identificazione (PII) come nomi, indirizzi, date, numeri di carta di credito e numeri di previdenza sociale provenienti da documenti contenuti in un bucket Amazon S3.

Se nel bucket sono presenti documenti che includono informazioni quali numeri di carta di credito o informazioni sul conto corrente, è possibile configurare la funzione Oscuramento di PII di S3 Object

Lambda per rilevare le informazioni personali e quindi restituire una copia di questi documenti in cui i tipi di entità PII sono redatti.

Per iniziare, è sufficiente implementare la seguente funzione Lambda nell'account in uso e aggiungere l'ARN della funzione nella configurazione del punto di accesso Lambda per oggetti.

Di seguito è riportato un esempio di ARN per questa funzione:

```
arn:aws:serverlessrepo:us-east-1:111122223333::applications/
ComprehendPiiRedactionS3ObjectLambda
```

Puoi aggiungere o visualizzare questa funzione su AWS Management Console utilizzando il seguente AWS Serverless Application Repository link: S3. ComprehendPiiRedaction ObjectLambda

Per visualizzare questa funzione su GitHub, consulta Amazon Comprehend S3 Object Lambda.

Per informazioni sulle end-to-end procedure complete per alcune attività di S3 Object Lambda nella redazione delle PII, consulta. Tutorial: rilevamento e oscuramento dei dati PII con S3 Object Lambda e Amazon Comprehend

Esempio 3: Decompressione

La funzione Lambda `S3ObjectLambdaDecompression` può decomprimere gli oggetti archiviati in Amazon S3 in uno dei sei formati di file compressi: `bzip2`, `gzip`, `snappy`, `zlib`, `zstandard` e `ZIP`.

Per iniziare, è sufficiente implementare la seguente funzione Lambda nell'account in uso e aggiungere l'ARN della funzione nella configurazione del punto di accesso Lambda per oggetti.

Di seguito è riportato un esempio di ARN per questa funzione:

```
arn:aws:serverlessrepo:us-east-1:111122223333::applications/S3ObjectLambdaDecompression
```

È possibile aggiungere o visualizzare questa funzione su AWS Management Console utilizzando il seguente AWS Serverless Application Repository link: S3. ObjectLambdaDecompression

Per visualizzare questa funzione attiva GitHub, consulta S3 Object Lambda Decompression.

Best practice e linee guida per S3 Object Lambda

Quando si utilizza S3 Object Lambda, segui queste best practice e linee guida per ottimizzare le operazioni e le prestazioni.

Argomenti

Utilizzo di S3 Object Lambda

S3 Object Lambda supporta solo l'elaborazione delle richieste `GET`, `LIST` e `HEAD`. Qualsiasi altra richiesta non viene richiamata AWS Lambda e restituisce invece risposte API standard non trasformate. È possibile creare un massimo di 1.000 punti di accesso Object Lambda Account AWS per regione. La AWS Lambda funzione da utilizzare deve trovarsi nella stessa Account AWS regione dell'Object Lambda Access Point.

S3 Object Lambda richiede fino a 60 secondi per trasmettere una risposta completa al suo chiamante. La tua funzione è inoltre soggetta a quote AWS Lambda predefinite. Per ulteriori informazioni, consulta la sezione Quote Lambda nella Guida per gli sviluppatori di AWS Lambda .

Quando S3 Object Lambda richiama la funzione Lambda specificata, è responsabilità dell'utente garantire che tutti i dati sovrascritti o eliminati in Amazon S3 dalla funzione Lambda o dall'applicazione specificata siano quelli desiderati e corretti.

S3 Object Lambda può essere utilizzato solo per eseguire operazioni sugli oggetti. Non è possibile utilizzarlo per eseguire altre operazioni Amazon S3, ad esempio la modifica o l'eliminazione dei bucket. Per un elenco completo delle operazioni S3 che supportano gli access point, consulta Compatibilità dei punti di accesso con le operazioni S3.

Oltre a questo elenco, i punti di accesso Lambda per oggetti non supportano le operazioni `POST Object`, `CopyObject` (come origine) e le operazioni API `SelectObjectContent`.

Servizi AWS utilizzato in combinazione con S3 Object Lambda

S3 Object Lambda collega Amazon S3 e AWS Lambda, facoltativamente, Servizi AWS altri di tua scelta per fornire oggetti pertinenti alle applicazioni richiedenti. Tutti i Servizi AWS dispositivi utilizzati con S3 Object Lambda sono regolati dai rispettivi Service Level Agreement (SLA). Ad esempio, se qualcuno Servizio AWS non rispetta il proprio impegno di servizio, hai diritto a ricevere un credito di servizio, come documentato nello SLA del servizio.

Intestazioni **`Range`** e **`partNumber`**

In caso di utilizzo di oggetti di grandi dimensioni, è possibile usare l'intestazione HTTP `Range` per scaricare un intervallo di byte specificato da un oggetto. Quando si utilizza l'intestazione `Range`, la richiesta recupera solo la parte specificata dell'oggetto. È anche possibile utilizzare l'intestazione `partNumber` per eseguire una richiesta basata su intervallo per la parte specificata dall'oggetto.

Per ulteriori informazioni, consulta Utilizzo delle intestazioni Range e partNumber.

Trasformazione di **`expiry-date`**

È possibile aprire o scaricare oggetti trasformati dal punto di accesso Object Lambda su. AWS Management Console Questi oggetti non devono essere scaduti. Se la funzione Lambda trasforma `expiry-date` degli oggetti, potrebbero venire visualizzati oggetti scaduti che non possono essere aperti o scaricati. Questo comportamento si applica solo agli oggetti ripristinati in S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive.

Utilizzo degli SDK AWS CLI e AWS

AWS Command Line Interface (AWS CLI) I sottocomandi S3 (`cp`,`mv`, and`sync`) e l'uso della AWS SDK for Java `TransferManager` classe non sono supportati per l'uso con S3 Object Lambda.

Tutorial di S3 Object Lambda

I seguenti tutorial presentano end-to-end procedure complete per alcune attività di S3 Object Lambda.

- Tutorial: trasformazione dei dati per l'applicazione con S3 Object Lambda
- Tutorial: rilevamento e oscuramento dei dati PII con S3 Object Lambda e Amazon Comprehend
- Tutorial: utilizzo di S3 Object Lambda per aggiungere filigrane alle immagini in modo dinamico man mano che vengono recuperate

Debug di S3 Object Lambda

Le richieste ai punti di accesso Amazon S3 Object Lambda possono comportare una nuova risposta di errore quando si verificano problemi con l'invocazione o l'esecuzione della funzione Lambda. Questi errori seguono lo stesso formato degli errori Amazon S3 standard. Per informazioni sugli errori di S3 Object Lambda, consulta la sezione Elenco dei codici di errore di S3 Object Lambda nei riferimenti all'API di Amazon Simple Storage Service.

Per ulteriori informazioni sul debug generale delle funzioni Lambda, consulta Monitoraggio e risoluzione dei problemi delle applicazioni Lambda nella Guida per gli sviluppatori di AWS Lambda .

Per informazioni sugli errori standard di Amazon S3, consulta la sezione Risposte agli errori nei riferimenti all'API di Amazon Simple Storage Service.

Puoi abilitare i parametri delle richieste in Amazon CloudWatch per i tuoi access point Object Lambda. Queste metriche possono essere utilizzate per monitorare le prestazioni operative del punto di accesso. È possibile abilitare le metriche delle richieste durante o dopo la creazione del punto di accesso Lambda per oggetti. Per ulteriori informazioni, consulta Metriche della richiesta S3 Object Lambda in CloudWatch.

Puoi abilitare gli eventi di dati AWS CloudTrail per ottenere una registrazione più granulare sulle richieste effettuate ai punti di accesso Lambda per oggetti. Per ulteriori informazioni, consulta Registrazione di eventi di dati per i percorsi nella Guida per l'utente di AWS CloudTrail .

Per i tutorial su S3 Object Lambda, consulta quanto segue:

- Tutorial: trasformazione dei dati per l'applicazione con S3 Object Lambda
- Tutorial: rilevamento e oscuramento dei dati PII con S3 Object Lambda e Amazon Comprehend
- Tutorial: utilizzo di S3 Object Lambda per aggiungere filigrane alle immagini in modo dinamico man mano che vengono recuperate

Per ulteriori informazioni sui punti di accesso standard, consulta la sezione Gestione dell'accesso ai dati con Punti di accesso Amazon S3.

Per informazioni sull'utilizzo di bucket, consulta Panoramica dei bucket. Per informazioni sull'utilizzo di oggetti, consulta Panoramica degli oggetti di Amazon S3.

Che cos'è S3 Express One Zone?

Amazon S3 Express One Zone è una classe di archiviazione Amazon S3 a zona singola ad alte prestazioni, creata appositamente per fornire un accesso ai dati coerente di pochi millisecondi per le applicazioni sensibili alla latenza. S3 Express One Zone è la classe di storage di oggetti cloud con la latenza più bassa disponibile oggi, con velocità di accesso ai dati fino a 10 volte più veloci e con costi di richiesta inferiori del 50% rispetto a S3 Standard. Le applicazioni possono trarre immediatamente vantaggio dal fatto che le richieste vengano completate fino a un ordine di grandezza più velocemente. S3 Express One Zone offre un'elasticità prestazionale simile a quella delle altre classi di storage S3.

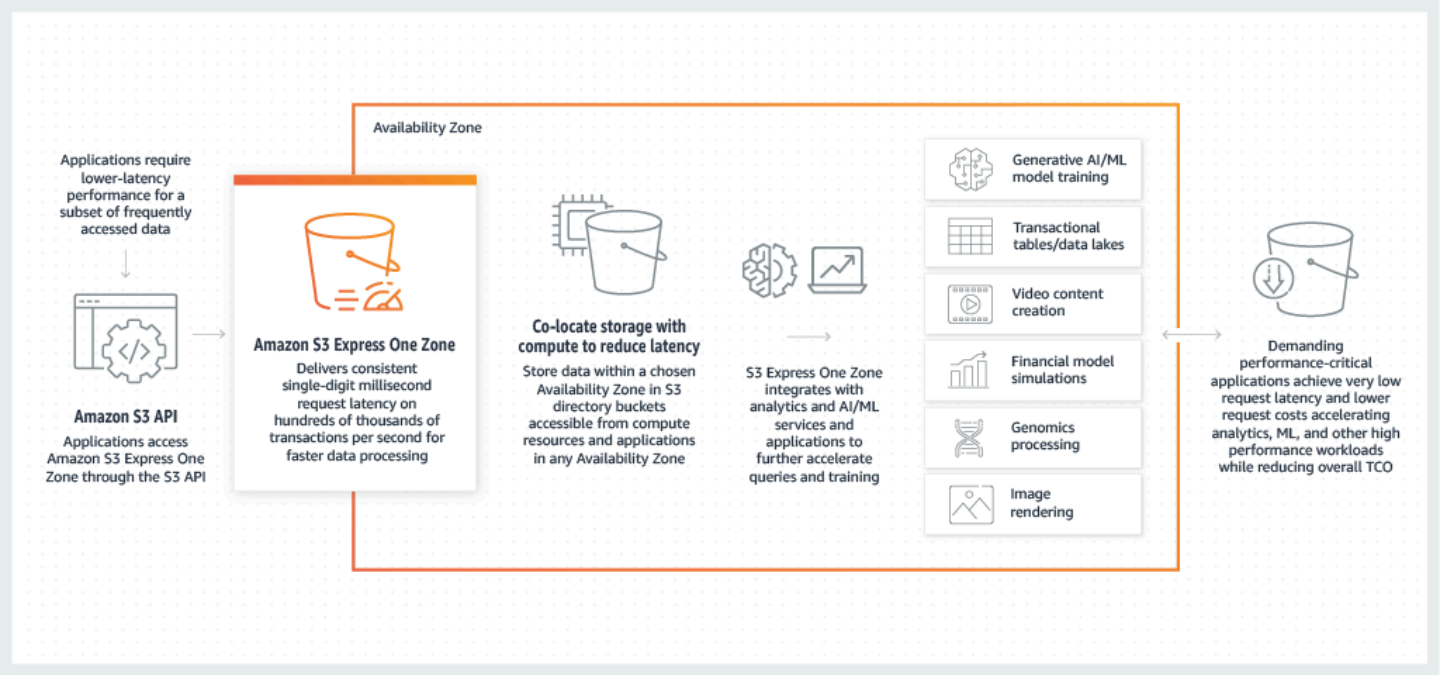
Come per le altre classi di storage di Amazon S3, non è necessario pianificare o fornire in anticipo i requisiti di capacità o throughput. Puoi aumentare o ridurre lo storage in base alle necessità e accedere ai dati tramite l'API Amazon S3.

S3 Express One Zone è la prima classe di archiviazione S3 in cui è possibile selezionare una singola zona di disponibilità con la possibilità di co-ubicare l'archiviazione di oggetti con le risorse di calcolo, che offre la massima velocità di accesso possibile. Inoltre, per aumentare ulteriormente la velocità di accesso e supportare centinaia di migliaia di richieste al secondo, i dati nella classe di storage S3 Express One Zone vengono archiviati in un nuovo tipo di bucket: un bucket di directory Amazon S3. Ogni bucket di directory può supportare centinaia di migliaia di transazioni al secondo (TPS), a prescindere dai nomi delle chiavi o dal modello di accesso.

La classe di storage Amazon S3 Express One Zone è progettata per una disponibilità del 99,95% all'interno di una singola zona di disponibilità ed è supportata dal Service Level Agreement di Amazon S3. Con S3 Express One Zone, i dati vengono archiviati in modo ridondante su più dispositivi all'interno di una singola zona di disponibilità. S3 Express One Zone è progettata per gestire guasti simultanei dei dispositivi rilevando e riparando rapidamente l'eventuale ridondanza persa. Se il dispositivo esistente rileva un guasto, S3 Express One Zone sposta automaticamente le richieste in nuovi dispositivi all'interno di una zona di disponibilità. Questa ridondanza garantisce l'accesso ininterrotto ai dati all'interno di una zona di disponibilità.

S3 Express One Zone è ideale per qualsiasi applicazione in cui è importante ridurre al minimo la latenza richiesta per accedere a un oggetto. Tali applicazioni possono essere flussi di lavoro interattivi con l'uomo, come l'editing video, in cui i professionisti creativi necessitano di un accesso reattivo ai contenuti dalle loro interfacce utente. S3 Express One Zone beneficia, inoltre, di carichi di lavoro di analisi e machine learning che hanno requisiti di reattività simili ai relativi dati, in particolare

carichi di lavoro con molti accessi più piccoli o un numero elevato di accessi casuali. S3 Express One Zone può essere utilizzato con altri Servizi AWS per supportare carichi di lavoro di analisi, intelligenza artificiale e machine learning (AI/ML), come Amazon EMR, SageMaker Amazon e Amazon Athena.



Quando usi S3 Express One Zone, puoi interagire con il tuo bucket di directory in un cloud privato virtuale (VPC) utilizzando un endpoint VPC gateway. Con un endpoint gateway, puoi accedere ai bucket di directory S3 Express One Zone dal tuo VPC senza un gateway Internet o un dispositivo NAT per il tuo VPC e senza costi aggiuntivi.

Puoi utilizzare molte delle stesse operazioni e funzionalità dell'API di Amazon S3 con i bucket di directory utilizzati con i bucket generici e altre classi di storage. Queste includono Mountpoint per Amazon S3, crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3), Operazioni in batch S3 e Blocco dell'accesso pubblico S3. Puoi accedere a S3 Express One Zone utilizzando la console Amazon S3 AWS Command Line Interface ,AWS CLI() AWS , gli SDK e l'API REST di Amazon S3.

Per ulteriori informazioni su S3 Express One Zone, consulta i seguenti argomenti.

- Panoramica
- Funzionalità di S3 Express One Zone
- Servizi correlati
- Passaggi successivi

Panoramica

Per ottimizzare le prestazioni e ridurre la latenza, S3 Express One Zone introduce i seguenti nuovi concetti.

Zona di disponibilità singola

La classe di storage Amazon S3 Express One Zone è progettata per una disponibilità del 99,95% all'interno di una singola zona di disponibilità ed è supportata dal Service Level Agreement di Amazon S3. Con S3 Express One Zone, i dati vengono archiviati in modo ridondante su più dispositivi all'interno di una singola zona di disponibilità. S3 Express One Zone è progettata per gestire guasti simultanei dei dispositivi rilevando e riparando rapidamente l'eventuale ridondanza persa. Se il dispositivo esistente rileva un guasto, S3 Express One Zone sposta automaticamente le richieste in nuovi dispositivi all'interno di una zona di disponibilità. Questa ridondanza garantisce l'accesso ininterrotto ai dati all'interno di una zona di disponibilità.

Una zona di disponibilità consiste in uno o più data center separati con alimentazione, rete e connettività ridondanti in una Regione AWS. Quando crei un bucket di directory, scegli la zona di disponibilità e Regione AWS dove collocare il bucket.

Bucket di directory

Esistono due tipi di bucket Amazon S3: i bucket generici S3 e i bucket di directory S3. I bucket per uso generico sono il tipo di bucket Amazon S3 predefinito utilizzato per la maggior parte dei casi d'uso S3. I bucket di directory utilizzano solo la classe di archiviazione S3 Express One Zone, progettata per carichi di lavoro o applicazioni con prestazioni critiche che richiedono una latenza costante di pochi millisecondi. Scegli il tipo di bucket più adatto alle tue esigenze applicative e prestazionali.

I bucket di directory organizzano i dati gerarchicamente in directory, a differenza della struttura di archiviazione piatta dei bucket generici. Non ci sono limiti di prefissi per i bucket di directory e le singole directory possono essere dimensionate orizzontalmente.

I bucket di directory utilizzano la classe di archiviazione S3 Express One Zone, creata per essere utilizzata da applicazioni sensibili alle prestazioni. Con S3 Express One Zone, puoi selezionare una singola zona di disponibilità con la possibilità di co-localizzare lo storage di oggetti con le tue risorse di elaborazione, il che offre la massima velocità di accesso possibile. Ciò è diverso dai bucket per uso generico, che archiviano oggetti in modo ridondante su più zone di disponibilità in. Regioni AWS

Per ulteriori informazioni sui bucket di directory, consulta Bucks di directory. Per ulteriori informazioni sui bucket per uso generico, consulta Panoramica dei bucket.

Endpoint ed endpoint VPC del gateway

Le operazioni API di gestione dei bucket per i bucket di directory sono disponibili tramite un endpoint regionale e sono denominate operazioni API degli endpoint regionali. Esempi di operazioni API degli endpoint regionali sono `CreateBucket` e `DeleteBucket`. Dopo aver creato un bucket di directory, puoi utilizzare le operazioni API degli endpoint zonali per caricare e gestire gli oggetti nel bucket di directory. Le operazioni API degli endpoint zonali sono disponibili tramite un endpoint zonale. Esempi di operazioni API degli endpoint zonali sono `PutObject` e `CopyObject`.

Puoi accedere a S3 Express One Zone dal tuo VPC utilizzando gli endpoint VPC gateway. Dopo aver creato un endpoint del gateway, puoi aggiungerlo come una destinazione nella tabella di routing per il traffico in transito dal VPC a S3 Express One Zone. Analogamente ad Amazon S3, l'utilizzo di endpoint del gateway non comporta costi supplementari. Per ulteriori informazioni su come configurare gli endpoint VPC del gateway, consulta Servizi di rete per S3 Express One Zone

Autorizzazione basata sulla sessione

Con S3 Express One Zone, autentichi e autorizzi le richieste tramite un nuovo meccanismo basato sulla sessione, ottimizzato per fornire la latenza più bassa. Puoi utilizzare `CreateSession` per richiedere credenziali temporanee che forniscono un accesso a bassa latenza al bucket. Queste credenziali temporanee sono definite per un bucket di directory S3 specifico. I token di sessione vengono utilizzati solo con operazioni zonali (a livello di oggetto) (ad eccezione di). CopyObject Per ulteriori informazioni, consulta Autorizzazione `CreateSession`.

Gli AWS SDK supportati per S3 Express One Zone gestiscono la creazione e l'aggiornamento delle sessioni per tuo conto. Per proteggere le sessioni, le credenziali di sicurezza temporanee scadono dopo 5 minuti. Dopo aver scaricato e installato gli AWS SDK e configurato le autorizzazioni AWS Identity and Access Management (IAM) necessarie, puoi iniziare immediatamente a utilizzare le operazioni API.

Funzionalità di S3 Express One Zone

Le seguenti funzionalità S3 sono disponibili per S3 Express One Zone. Per un elenco completo delle operazioni API supportate e delle funzionalità non supportate, consulta. In cosa differisce S3 Express One Zone?

Gestione degli accessi e sicurezza

Con i bucket di directory, puoi utilizzare le seguenti funzionalità per eseguire l'audit e gestire l'accesso. Per impostazione predefinita, i bucket di directory sono privati e l'accesso è possibile solo dagli utenti a cui è concesso esplicitamente l'accesso. A differenza dei bucket per uso generico, che possono impostare il limite di controllo dell'accesso a livello di bucket, prefisso o tag dell'oggetto, il limite di controllo dell'accesso per i bucket di directory viene impostato solo a livello di bucket. Per ulteriori informazioni, consulta AWS Identity and Access Management (IAM) per S3 Express One Zone.

- S3 Block Public Access: tutte le impostazioni di S3 Block Public Access sono abilitate per impostazione predefinita a livello di bucket. Questa impostazione predefinita non può essere modificata.
- S3 Object Ownership (proprietario del bucket applicato per impostazione predefinita): le liste di controllo degli accessi (ACL) non sono supportate per i bucket di directory. I bucket di directory utilizzano automaticamente l'impostazione imposta dal proprietario del bucket per S3 Object Ownership. L'applicazione del proprietario del bucket significa che gli ACL sono disabilitati e il proprietario del bucket possiede automaticamente e ha il pieno controllo su ogni oggetto nel bucket. Questa impostazione predefinita non può essere modificata.
- AWS Identity and Access Management (IAM): IAM ti aiuta a controllare in modo sicuro l'accesso ai tuoi bucket di directory. Puoi utilizzare IAM per concedere l'accesso alle operazioni API di gestione dei bucket (regionali) e alle operazioni API di gestione degli oggetti (zonali) tramite l'azione. `s3express:CreateSession` Per ulteriori informazioni, consulta AWS Identity and Access Management (IAM) per S3 Express One Zone. A differenza delle azioni di gestione degli oggetti, le azioni di gestione dei bucket non possono essere multi-account. Solo il proprietario del bucket può eseguire tali azioni.
- Policy di bucket: utilizza il linguaggio delle policy basato su IAM per configurare le autorizzazioni basate sulle risorse per i bucket di directory. Puoi anche utilizzare IAM per controllare l'accesso al funzionamento dell'`CreateSession`API, il che ti consente di utilizzare le operazioni API Zonal o di gestione degli oggetti. Puoi concedere l'accesso allo stesso account o a più account alle operazioni dell'API Zonal. Per ulteriori informazioni sulle autorizzazioni e le politiche di S3 Express One Zone, consulta. AWS Identity and Access Management (IAM) per S3 Express One Zone
- IAM Access Analyzer for S3: valuta e monitora le tue policy di accesso per assicurarti che forniscano solo l'accesso previsto alle tue risorse S3.

Registrazione di log e monitoraggio

S3 Express One Zone utilizza i seguenti strumenti di registrazione e monitoraggio S3 che puoi utilizzare per monitorare e controllare il modo in cui vengono utilizzate le tue risorse:

- CloudWatch Parametri Amazon: monitora AWS le tue risorse e le tue applicazioni utilizzandole CloudWatch per raccogliere e tenere traccia dei parametri. S3 Express One Zone utilizza lo stesso spazio dei CloudWatch nomi delle altre classi di storage Amazon S3 (`AWS/S3`) e supporta i parametri di storage giornalieri per i bucket di directory: e. `BucketSizeBytes` `NumberOfObjects` Per ulteriori informazioni, consulta Monitoraggio delle metriche con Amazon CloudWatch.
- AWS CloudTrail logs: AWS CloudTrail è uno strumento Servizio AWS che ti aiuta a implementare il controllo operativo e dei rischi, la governance e la conformità della tua azienda registrando le azioni intraprese Account AWS da un utente, ruolo o un. Servizio AWS Per S3 Express One Zone, CloudTrail acquisisce le operazioni delle API degli endpoint regionali (ad esempio, `CreateBucket` e`PutBucketPolicy`) come eventi di gestione. Questi eventi includono le azioni intraprese nelle operazioni AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDK e API. AWS Gli eventi `eventsource` per la CloudTrail gestione di S3 Express One Zone sono. `s3express.amazonaws.com` Per ulteriori informazioni, consulta Eventi Amazon S3 CloudTrail .

> **Note**
>
> I log di accesso al server Amazon S3 non sono supportati con S3 Express One Zone.

Gestione degli oggetti

Dopo aver creato un bucket di directory, puoi gestire lo storage di oggetti utilizzando la console Amazon S3 AWS , gli SDK e. AWS CLI Le seguenti funzionalità sono disponibili per la gestione degli oggetti con S3 Express One Zone:

- Operazioni Batch S3: utilizza le operazioni batch per eseguire operazioni in blocco sugli oggetti nei bucket di directory, ad esempio la funzione Copy and Invoke. AWS Lambda Ad esempio, puoi utilizzare Operazioni in batch per copiare oggetti tra bucket di directory e bucket per uso generico. Con Batch Operations, puoi gestire miliardi di oggetti su larga scala con una singola richiesta S3 utilizzando gli AWS SDK AWS CLI o con pochi clic nella console Amazon S3.

- Importa: dopo aver creato un bucket di directory, puoi popolarlo con oggetti utilizzando la funzionalità di importazione nella console Amazon S3. L'importazione è un metodo ottimizzato di creazione di processi Operazioni in batch per copiare oggetti da bucket per uso generico in bucket di directory.

AWS SDK e librerie client

Dopo aver creato un bucket di directory e caricato un oggetto nel bucket, puoi gestire l'archiviazione degli oggetti utilizzando quanto segue.

- Mountpoint per Amazon S3 — Mountpoint per Amazon S3 è un client di file open source che offre un accesso a throughput elevato, riducendo i costi di elaborazione per i data lake su Amazon S3. Mountpoint per Amazon S3 traduce le chiamate API del file system locale in chiamate API di oggetti S3 come e. `GET` `LIST` È ideale per carichi di lavoro di data lake ad alta intensità di lettura che elaborano petabyte di dati e richiedono l'elevata velocità di trasmissione elastica fornita da Amazon S3 per scalare verso l'alto e verso il basso su migliaia di istanze.
- S3A— S3A è un'interfaccia Hadoop compatibile consigliata per l'accesso agli archivi dati in Amazon S3. S3Asostituisce il client del S3N Hadoop file system.
- PyTorchon AWS — PyTorch on AWS è un framework open source di deep learning che semplifica lo sviluppo di modelli di machine learning e la loro implementazione in produzione.
- AWS SDK: puoi utilizzare gli AWS SDK per sviluppare applicazioni con Amazon S3. Gli AWS SDK semplificano le attività di programmazione integrando l'API REST di Amazon S3 sottostante. Per ulteriori informazioni sull'utilizzo degli AWS SDK con S3 Express One Zone, consulta. the section called “AWS SDK”

Crittografia e protezione dei dati

Gli oggetti archiviati nei bucket di directory vengono crittografati automaticamente utilizzando la crittografia lato server con chiavi gestite di Amazon S3 (SSE-S3). I bucket di directory non supportano la crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS), la crittografia lato server con chiavi di crittografia fornite dal cliente (SSE-C) o la crittografia lato server a doppio livello con (DSSE-KMS). AWS KMS keys Per ulteriori informazioni, consulta Protezione e crittografia dei dati e Uso della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).

S3 Express One Zone offre la possibilità di scegliere l'algoritmo di checksum utilizzato per convalidare i dati durante il caricamento o il download. Puoi selezionare uno dei seguenti algoritmi

di controllo dell'integrità dei dati Secure Hash Algorithms (SHA) o Cyclic Redundancy Check (CRC): CRC32, CRC32C, SHA-1 e SHA-256. I checksum basati su MD5 non sono supportati con la classe di storage S3 Express One Zone.

Per ulteriori informazioni, consulta Best practice per il checksum S3 aggiuntivo.

AWS Versione Signature 4 () SigV4

S3 Express One Zone utilizza AWS la versione Signature 4 (SigV4). SigV4è un protocollo di firma utilizzato per autenticare le richieste ad Amazon S3 tramite HTTPS. S3 Express One Zone firma le richieste utilizzando. AWS Sigv4 Per ulteriori informazioni, consulta Authenticating Requests (AWS Signature Version 4) nel riferimento all'API di Amazon Simple Storage Service.

Forte coerenza

S3 Express One Zone offre una forte read-after-write coerenza per `DELETE` tutte `PUT` le richieste di oggetti presenti nei bucket di directory. Regioni AWS Per ulteriori informazioni, consulta Modello di consistenza dati Amazon S3.

Servizi correlati

Puoi utilizzare quanto segue Servizi AWS con la classe di storage S3 Express One Zone per supportare il tuo caso d'uso specifico a bassa latenza.

- Amazon Elastic Compute Cloud (Amazon EC2) — Amazon EC2 fornisce capacità di elaborazione sicura e scalabile in. Cloud AWS L'utilizzo Amazon EC2 riduce la necessità di investimenti anticipati in hardware e ti permette di sviluppare e distribuire più rapidamente le applicazioni. Puoi utilizzare Amazon EC2 per avviare il numero di server virtuali necessari, configurare la sicurezza e i servizi di rete, nonché gestire l'archiviazione.
- AWS Lambda: Lambda è un servizio di calcolo che consente di eseguire il codice senza provisioning o gestire server. È possibile configurare le impostazioni di notifica su un bucket e concedere ad Amazon S3 l'autorizzazione a invocare una funzione sulla policy di autorizzazione basata sulle risorse della funzione.
- Amazon Elastic Kubernetes Service (Amazon EKS) — Amazon EKS è un servizio gestito che elimina la necessità di installare, utilizzare e mantenere Kubernetes il proprio piano di controllo. AWSKubernetesè un sistema open source che automatizza la gestione, la scalabilità e la distribuzione di applicazioni containerizzate.

- Amazon Elastic Container Service (Amazon ECS): Amazon ECS è un servizio di orchestrazione di container completamente gestito che facilita l'implementazione, la gestione e il dimensionamento di applicazioni distribuite in container.
- Amazon Athena: Athena è un servizio di query interattivo che semplifica l'analisi dei dati direttamente in Amazon S3 utilizzando SQL standard. Puoi anche utilizzare Athena per eseguire analisi dei dati in modo interattivo Apache Spark senza dover pianificare, configurare o gestire le risorse. Quando esegui Apache Spark applicazioni su Athena, invii il Spark codice per l'elaborazione e ricevi direttamente i risultati.
- Amazon SageMaker Runtime Model Training — Amazon SageMaker Runtime è un servizio di machine learning completamente gestito. Con SageMaker Runtime, data scientist e sviluppatori possono creare e addestrare modelli di machine learning in modo rapido e semplice e poi distribuirli direttamente in un ambiente ospitato pronto per la produzione.
- AWS Glue— AWS Glue è un servizio di integrazione dei dati senza server che consente agli utenti di analisi di scoprire, preparare, spostare e integrare facilmente i dati provenienti da più fonti. È possibile utilizzarlo AWS Glue per l'analisi, l'apprendimento automatico e lo sviluppo di applicazioni. AWS Glue include anche strumenti di produttività e data-ops aggiuntivi per la creazione, l'esecuzione di lavori e l'implementazione dei flussi di lavoro aziendali.
- Amazon EMR: Amazon EMR è una piattaforma di cluster gestita che semplifica l'esecuzione di framework di big data, come «and on», AWS per elaborare Apache Hadoop e analizzare Apache Spark grandi quantità di dati.

Passaggi successivi

Per ulteriori informazioni sull'utilizzo della classe di archiviazione S3 Express One Zone e dei bucket di directory, consulta gli argomenti seguenti:

- In cosa differisce S3 Express One Zone?
- Nozioni di base su S3 Express One Zone
- Servizi di rete per S3 Express One Zone
- Bucks di directory
- Lavorare con oggetti in un bucket di directory
- Sicurezza per S3 Express One Zone
- Ottimizzazione delle prestazioni di Amazon S3 Express One Zone
- Sviluppo con S3 Express One Zone

In cosa differisce S3 Express One Zone?

Amazon S3 Express One Zone è una classe di archiviazione Amazon S3 a zona singola ad alte prestazioni, creata appositamente per fornire un accesso ai dati coerente di pochi millisecondi per le applicazioni sensibili alla latenza. S3 Express One Zone è la prima classe di archiviazione S3 in cui è possibile selezionare una singola zona di disponibilità con la possibilità di co-ubicare l'archiviazione di oggetti con le risorse di calcolo, che offre la massima velocità di accesso possibile. Inoltre, per aumentare ulteriormente la velocità di accesso e supportare centinaia di migliaia di richieste al secondo, i dati di S3 Express One Zone vengono archiviati in un nuovo tipo di bucket: un bucket di directory Amazon S3.

Per ulteriori informazioni, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Puoi creare bucket di directory e accedere ai dati in S3 Express One Zone mediante l'API Amazon S3. L'API Amazon S3 è compatibile con S3 Express One Zone e i bucket di directory, fatta eccezione per alcune differenze significative. Per ulteriori informazioni su come differisce S3 Express One Zone, consulta i seguenti argomenti.

Argomenti

- Differenze di S3 Express One Zone
- Operazioni API supportate da S3 Express One Zone
- Funzionalità Amazon S3 non supportate da S3 Express One Zone

Differenze di S3 Express One Zone

- Tipo di bucket supportato: gli oggetti nella classe di archiviazione S3 Express One Zone possono essere archiviati solo in bucket di directory. Per ulteriori informazioni, consulta Bucks di directory.
- Durabilità: con S3 Express One Zone, i dati vengono archiviati in modo ridondante su più dispositivi all'interno di una singola zona di disponibilità. S3 Express One Zone è progettato per una disponibilità del 99,95% all'interno di una singola zona di disponibilità ed è supportato dall'Accordo sul livello di servizio di Amazon S3. Per ulteriori informazioni, consulta Zona di disponibilità singola.
- **`ListObjectsV2`**comportamento
  - Per i bucket di directory, `ListObjectsV2` non restituisce gli oggetti in ordine lessicografico (alfabetico). Inoltre, i prefissi devono terminare con un delimitatore che può corrispondere solo a "/".

  - Per i bucket di directory, `ListObjectsV2` response include i prefissi correlati solo ai caricamenti multiparte in corso.
- Comportamento di eliminazione: quando si elimina un oggetto in un bucket di directory, Amazon S3 elimina in modo ricorsivo tutte le directory vuote nel percorso dell'oggetto. Ad esempio, se elimini la chiave dell'oggetto`dir1/dir2/file1.txt`, Amazon S3 la elimina. `file1.txt` Se le directory `dir1/` e `dir2/` sono vuote e non contengono altri oggetti, Amazon S3 elimina anche tali directory.
- ETag e checksum: i tag di entità (ETag) per S3 Express One Zone sono stringhe alfanumeriche casuali e non checksum MD5. Per ulteriori informazioni sull'utilizzo di checksum aggiuntivi con S3 Express One Zone, consulta Best practice per il checksum S3 aggiuntivo.
- Chiavi degli oggetti nelle richieste **`DeleteObjects`**
  - Le chiavi degli oggetti nelle richieste `DeleteObjects` devono contenere almeno un carattere diverso dallo spazio. Le stringhe con tutti caratteri spaziatura non sono supportate nelle richieste `DeleteObjects`.
  - Le chiavi degli oggetti nelle richieste `DeleteObjects` non possono contenere caratteri di controllo Unicode, fatta eccezione per newline (`\n`) tab (`\t`) e carriage feed (`\r`).
- Endpoint regionali e zonali: quando si utilizza S3 Express One Zone, è necessario specificare la regione in tutte le richieste client. Per gli endpoint regionali, si specifica la regione, ad esempio `s3express-control.us-west-2.amazonaws.com`. Per gli endpoint zonali, si specificano la regione e la zona di disponibilità, ad esempio `s3express-usw2-az1.us-west-2.amazonaws.com`. Per ulteriori informazioni, consulta Endpoint regionali e zonali.
- Caricamenti in più parti: come per altri oggetti archiviati in Amazon S3, è possibile caricare e copiare oggetti di grandi dimensioni archiviati nella classe di archiviazione S3 Express One Zone utilizzando il processo di caricamento in più parti. Tuttavia, di seguito sono riportate alcune differenze quando si utilizza il processo di caricamento in più parti con oggetti archiviati in S3 Express One Zone. Per ulteriori informazioni, consulta the section called “Utilizzo di caricamenti multiparte con bucket di directory”.
  - L'ora di creazione dell'oggetto è la data di completamento del caricamento in più parti.
  - I numeri parte in più parti devono utilizzare numeri parte consecutivi. Se si tenta di completare una richiesta caricamento in più parti con numeri parte non consecutivi, Amazon S3 genera un errore `400 (Bad Request)` HTTP.
  - L'iniziatore di un caricamento in più parti può interrompere la richiesta di caricamento in più parti solo se è stata concesso esplicitamente l'accesso a `AbortMultipartUpload` tramite l'autorizzazione `s3express:CreateSession`. Per ulteriori informazioni, consulta AWS Identity and Access Management (IAM) per S3 Express One Zone.

- Svuotamento di un bucket di directory: il `s3 rm` comando tramite (AWS Command Line Interface CLI), l'operazione tramite Mountpoint e il pulsante di opzione Svuota il bucket tramite (CLI), l'`delete`operazione tramite Mountpoint e il pulsante di opzione Svuotamento del bucket di directory non AWS Management Console sono in grado di eliminare i caricamenti multiparte in corso in un bucket di directory. Per eliminare questi caricamenti multiparte in corso, utilizzate l'`ListMultipartUploads`operazione per elencare i caricamenti multiparte in corso nel bucket e utilizzate l'operazione per interrompere tutti i caricamenti multiparte in corso. `AbortMultupartUpload`

Operazioni API supportate da S3 Express One Zone

La classe di archiviazione Amazon S3 Express One Zone supporta operazioni API degli endpoint regionali (a livello di bucket o piano di controllo (control-plane)) e zonali (a livello di oggetto o piano dati). Per ulteriori informazioni, consulta Servizi di rete per S3 Express One Zone e Endpoint ed endpoint VPC del gateway.

Operazioni API degli endpoint regionali

Le seguenti operazioni API degli endpoint regionali sono supportate per S3 Express One Zone:

- CreateBucket
- DeleteBucket
- DeleteBucketPolicy
- GetBucketPolicy
- ListDirectoryBuckets
- PutBucketPolicy

Operazioni API degli endpoint zonali

Le seguenti operazioni API degli endpoint zonali sono supportate per S3 Express One Zone:

- CreateSession
- CopyObject
- DeleteObject
- DeleteObjects
- GetObject

- GetObjectAttributes
- HeadBucket
- HeadObject
- ListObjectsV2
- PutObject
- AbortMultipartUpload
- CompleteMultiPartUpload
- CreateMultipartUpload
- ListMultipartUploads
- ListParts
- UploadPart
- UploadPartCopy

Funzionalità Amazon S3 non supportate da S3 Express One Zone

Le seguenti funzionalità Amazon S3 non sono supportate da S3 Express One Zone:

- AWS CloudTrail eventi sul piano dati
- AWS politiche gestite
- AWS PrivateLink per S3
- Checksum MD5
- Eliminazione dell'autenticazione a più fattori (MFA)
- Blocco di oggetti in S3
- pagamento a carico del richiedente
- S3 Access Grants
- Punto di accesso S3
- Etichette bucket
- Metriche delle CloudWatch richieste Amazon
- Notifiche di eventi di Amazon S3
- Ciclo di vita S3
- Punti di accesso multi-regione S3
- Punti di accesso Lambda per oggetti S3

- Funzione Controllo delle versioni S3
- Inventario S3
- Replica di Amazon S3
- Tag oggetti
- S3 Select
- Log di accesso al server
- Hosting di siti Web statici
- S3 Storage Lens
- Gruppi Storage Lens S3
- Transfer Acceleration S3
- Crittografia lato server a doppio livello con AWS Key Management Service () chiavi (AWS KMS DSSE-KMS)
- Crittografia lato server con chiavi () (SSE-KMS) AWS Key Management Service AWS KMS
- Crittografia lato server con chiavi fornite dal cliente (SSE-C)
- L'opzione per copiare le impostazioni di un bucket esistente quando si crea un nuovo bucket. AWS Management Console

Nozioni di base su S3 Express One Zone

Nella sezione seguente viene descritto come iniziare a utilizzare la classe di archiviazione Amazon S3 Express One Zone e i bucket di directory. Per ulteriori informazioni, consulta Che cos'è S3 Express One Zone?.

Argomenti

- Configurazione AWS Identity and Access Management (IAM) con S3 Express One Zone
- Configurazione degli endpoint VPC del gateway
- Lavora con S3 Express One Zone utilizzando la console S3 e gli SDK AWS CLIAWS

Configurazione AWS Identity and Access Management (IAM) con S3 Express One Zone

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta gli amministratori a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM

controllano chi può essere autenticato (accesso effettuato) e autorizzato (dotato di autorizzazioni) a utilizzare le risorse Amazon S3 in S3 Express One Zone. Puoi utilizzare IAM senza alcun costo aggiuntivo.

Per impostazione predefinita, gli utenti non dispongono delle autorizzazioni per i bucket di directory e le operazioni S3 Express One Zone. Per concedere le autorizzazioni di accesso per i bucket di directory e le operazioni S3 Express One Zone, puoi utilizzare IAM per creare utenti o ruoli e collegare autorizzazioni a tali identità.

Per iniziare a utilizzare IAM, consulta AWS Identity and Access Management (IAM) per S3 Express One Zone e Policy basate sulle identità IAM per S3 Express One Zone.

Configurazione degli endpoint VPC del gateway

Per accedere a S3 Express One Zone, si utilizzano endpoint regionali e zonali diversi dagli endpoint Amazon S3 standard. A seconda dell'operazione API Amazon S3 utilizzata, è richiesto un endpoint regionale o zonale. Per un elenco completo delle operazioni API supportate per tipo di endpoint, consulta Operazioni API supportate da S3 Express One Zone. È necessario accedere agli endpoint regionali e zonali tramite un endpoint del cloud privato virtuale (VPC) del gateway. Per configurare gli endpoint del gateway, consulta Servizi di rete per S3 Express One Zone.

Lavora con S3 Express One Zone utilizzando la console S3 e gli SDK AWS CLIAWS

Puoi lavorare con la classe di storage S3 Express One Zone e i bucket di directory utilizzando gli AWS SDK, la console Amazon S3, AWS Command Line Interface ()AWS CLI e l'API REST di Amazon S3.

Console S3

Per iniziare a utilizzare la console S3, completa la procedura seguente:

- Creazione di un bucket di directory
- Svuotamento di un bucket di directory
- Eliminazione di un bucket di directory

AWS SDK

S3 Express One Zone supporta i seguenti AWS SDK:

- 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
- SDK AWS for Kotlin
- AWS SDK for Rust

Quando si utilizza S3 Express One Zone, si consiglia di utilizzare la versione più recente degli SDK AWS . Gli AWS SDK supportati per S3 Express One Zone gestiscono la creazione, l'aggiornamento e la chiusura della sessione per tuo conto. Ciò significa che puoi iniziare immediatamente a utilizzare le operazioni API dopo aver scaricato e installato gli AWS SDK e configurato le autorizzazioni IAM necessarie. Per ulteriori informazioni, consulta AWS Identity and Access Management (IAM) per S3 Express One Zone.

Per informazioni sugli AWS SDK, incluso come scaricarli e installarli, consulta Tools to Build on. AWS

Per esempi di AWS SDK, consulta quanto segue:

- Creazione di un bucket di directory
- Svuotamento di un bucket di directory
- Eliminazione di un bucket di directory

AWS Command Line Interface (AWS CLI)

Puoi usare AWS Command Line Interface (AWS CLI) per creare bucket di directory e utilizzare le operazioni API degli endpoint regionali e zonali supportate per S3 Express One Zone.

Per iniziare con AWS CLI, consulta Get started with the AWS CLI nel Command Reference.AWS CLI

> **Note**
>
> Per utilizzare i bucket di directory con i `aws s3` comandi di alto livello, aggiorna il tuo AWS CLI alla versione più recente. Per ulteriori informazioni su come installare e configurare AWS CLI, consulta Installare o aggiornare la versione più recente di AWS CLI nel AWS CLI Command Reference.

Per AWS CLI alcuni esempi, vedi quanto segue:

- Creazione di un bucket di directory
- Svuotamento di un bucket di directory
- Eliminazione di un bucket di directory

Servizi di rete per S3 Express One Zone

Per accedere agli oggetti della classe di archiviazione S3 Express One Zone e ai bucket di directory, si utilizzano endpoint API regionali e zonali diversi dagli endpoint Amazon S3 standard. A seconda dell'operazione API S3 utilizzata, è richiesto un endpoint regionale o zonale. Per un elenco completo di operazioni API per tipo di endpoint, consulta Operazioni API supportate da S3 Express One Zone.

Puoi accedere alle operazioni API regionali e zonali tramite gli endpoint del cloud privato virtuale (VPC) del gateway. Per configurare gli endpoint VPC del gateway, consulta the section called "Configurazione degli endpoint VPC del gateway".

Nei seguenti argomenti vengono descritti i requisiti di rete per accedere a S3 Express One Zone mediante un endpoint VPC del gateway.

Argomenti

- Endpoints
- Configurazione degli endpoint VPC del gateway

Endpoints

Puoi accedere agli oggetti della classe di archiviazione Amazon S3 Express One Zone e ai bucket di directory dal VPC mediante endpoint VPC del gateway. S3 Express One Zone utilizza endpoint

API regionali e zonali. A seconda dell'operazione API Amazon S3 utilizzata, è necessario un endpoint regionale o zonale. L'utilizzo di endpoint gateway non comporta costi supplementari.

Le operazioni API a livello di bucket (o piano di controllo (control-plane)) sono disponibili tramite endpoint regionali e sono denominate operazioni API degli endpoint regionali. Esempi di operazioni API degli endpoint regionali sono `CreateBucket` e `DeleteBucket`. Quando si crea un bucket di directory, si sceglie una singola zona disponibilità in cui verrà creato il bucket di directory. Dopo aver creato un bucket di directory, puoi utilizzare le operazioni API degli endpoint zonali per caricare e gestire gli oggetti nel bucket di directory.

Le operazioni API a livello di oggetto (o piano dati) sono disponibili tramite endpoint zonali e sono denominate operazioni API degli endpoint zonali. Esempi di operazioni API degli endpoint zonali sono `CreateSession` e `PutObject`.

Nella tabella seguente vengono mostrati gli endpoint API regionali e zonali disponibili per ogni regione e zona di disponibilità.

Configurazione degli endpoint VPC del gateway

Utilizza la procedura seguente per creare un endpoint del gateway che si connette agli oggetti della classe di archiviazione Amazon S3 Express One Zone e ai bucket di directory.

Per configurare gli endpoint VPC del gateway

1. Apri la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Nel pannello di navigazione, seleziona Endpoints (Endpoint).
3. Seleziona Crea endpoint.
4. Crea un nome per l'endpoint.
5. Per Service category (Categoria servizio), scegli Servizi AWS.
6. Per Servizi, aggiungi il filtro Type=Gateway, quindi scegli il pulsante di opzione accanto a com.amazonaws.***region***.s3express.
7. Per VPC, scegli un VPC in cui creare l'endpoint.
8. In Route tables (Tabelle di instradamento), seleziona le tabelle di instradamento che devono essere utilizzate dall'endpoint. Amazon VPC aggiunge automaticamente una route che indirizza il traffico destinato per il servizio all'interfaccia di rete dell'endpoint.
9. Per Policy, scegli Accesso completo per consentire a tutti i principali di eseguire tutte le operazioni su tutte le risorse dell'endpoint VPC. In caso contrario, scegli Personalizza per

collegare una policy dell'endpoint VPC che controlla le autorizzazioni di cui dispongono i principali per eseguire azioni sulle risorse dell'endpoint VPC.

10. (Facoltativo) Per aggiungere un tag, scegli Aggiungi nuovo tag e immetti la chiave e il valore del tag.
11. Seleziona Crea endpoint.

Dopo aver creato un endpoint del gateway, puoi utilizzare gli endpoint API regionali e gli endpoint API zonali per accedere agli oggetti della classe di archiviazione Amazon S3 Express One Zone e ai bucket di directory.

Bucks di directory

Esistono due tipi di bucket Amazon S3: i bucket per uso generico e i bucket di directory. Scegli il tipo di bucket più adatto ai requisiti applicativi e di prestazioni:

- I bucket per uso generico sono il tipo di bucket S3 originale e sono consigliati per la maggior parte dei casi d'uso e dei modelli di accesso. I bucket per uso generico consentono l'uso di oggetti archiviati in tutte le classi di archiviazione, fatta eccezione per S3 Express One Zone.
- I bucket di directory utilizzano la classe di archiviazione S3 Express One Zone, consigliata se l'applicazione è sensibile alle prestazioni e ottiene vantaggi da latenze `PUT` e `GET` di pochi millisecondi.

I bucket di directory vengono utilizzati per carichi di lavoro o applicazioni critiche per le prestazioni che richiedono una latenza costante di pochi millisecondi. I bucket di directory organizzano i dati in modo gerarchico in directory, a differenza della struttura di archiviazione piatta dei bucket per uso generico. Non ci sono limiti di prefissi per i bucket di directory e le singole directory possono essere dimensionate orizzontalmente.

I bucket di directory utilizzano la classe di archiviazione S3 Express One Zone, che archivia i dati in più dispositivi all'interno di una singola zona di disponibilità, ma non archivia i dati in modo ridondante nelle zone di disponibilità. Quando crei un bucket di directory, ti consigliamo di specificare una Regione AWS e una zona di disponibilità locale per le tue istanze di calcolo Amazon EC2, Amazon Elastic Kubernetes Service o Amazon Elastic Container Service (Amazon ECS) per ottimizzare le prestazioni.

Puoi creare fino a 10 bucket di directory in ciascuno dei tuoi Account AWS, senza limiti al numero di oggetti che puoi archiviare in un bucket. La quota del bucket viene applicata a ciascuna regione nell'

Account AWS. Se la tua applicazione richiede l'aumento di questo limite, contatta. AWS Support Per ulteriori informazioni, visita la console Service Quotas.

> **Important**
>
> I bucket di directory che non presentano alcuna attività di richiesta per un periodo di almeno 90 giorni passano a uno stato inattivo. In uno stato inattivo, un bucket di directory è temporaneamente inaccessibile per letture e scritture. I bucket inattivi mantengono tutta l'archiviazione, i metadati degli oggetti e i metadati dei bucket. I costi di storage esistenti si applicano ai bucket inattivi. Se effettui una richiesta di accesso a un bucket inattivo, il bucket passa a uno stato attivo, in genere entro pochi minuti. Durante questo periodo di transizione, le letture e le scritture restituiscono un codice di errore HTTP. `503 (Service Unavailable)`

I seguenti argomenti forniscono informazioni sui bucket di directory. Per ulteriori informazioni sui bucket per uso generico, consulta Panoramica dei bucket.

Argomenti

Zone di disponibilità

Quando si crea un bucket di directory, si sceglie la zona di disponibilità e la Regione AWS.

I bucket di directory utilizzano la classe di archiviazione S3 Express One Zone, creata per essere utilizzata da applicazioni sensibili alle prestazioni. S3 Express One Zone è la prima classe di archiviazione S3 in cui è possibile selezionare una singola zona di disponibilità con la possibilità di co-ubicare l'archiviazione di oggetti con le risorse di calcolo, che offre la massima velocità di accesso possibile.

Con S3 Express One Zone, i dati vengono archiviati in modo ridondante su più dispositivi all'interno di una singola zona di disponibilità. S3 Express One Zone è progettato per una disponibilità del 99,95% all'interno di una singola zona di disponibilità ed è supportato dal Service Level Agreement di Amazon S3. Per ulteriori informazioni, consulta Zona di disponibilità singola

Nomi dei bucket di directory

Il nome di un bucket di directory è costituito da un nome di base fornito dall'utente e da un suffisso contenente l'ID della zona di disponibilità in cui si trova il bucket. I nomi dei bucket di directory devono utilizzare il seguente formato e rispettare le regole di denominazione dei bucket di directory:

```
bucket-base-name--azid--x-s3
```

Ad esempio, il seguente nome del bucket di directory contiene l'ID zona di disponibilità `usw2-az1`:

```
bucket-base-name--usw2-az1--x-s3
```

Per ulteriori informazioni, consulta Regole di denominazione dei bucket di directory.

Directory

I bucket di directory organizzano i dati in modo gerarchico in directory, a differenza della struttura di ordinamento piatta dei bucket per uso generico. Ogni bucket di directory S3 può supportare centinaia di migliaia di transazioni al secondo (TPS), a prescindere dal numero di directory all'interno del bucket.

Con uno spazio dei nomi gerarchico, il delimitatore nella chiave dell'oggetto è importante. Il solo delimitatore supportato è una barra (`/`). Le directory sono determinate dai limiti dei delimitatori. Ad

esempio, la chiave dell'oggetto `dir1/dir2/file1.txt` comporta che le directory `dir1/` e `dir2/` vengano create automaticamente e che l'oggetto `file1.txt` venga aggiunto alla directory `/dir2` nel percorso `dir1/dir2/file1.txt`.

Il modello di indicizzazione del bucket di directory restituisce risultati non ordinati per l'operazione API `ListObjectsV2`. Se è necessario limitare i risultati a una sottosezione del bucket, è possibile specificare un percorso di sottodirectory nel parametro `prefix`, ad esempio `prefix=dir1/`.

Nomi delle chiavi

Per i bucket di directory, le sottodirectory comuni a più chiavi oggetto vengono create con la prima chiave dell'oggetto. Le chiavi oggetto aggiuntive per la stessa sottodirectory utilizzano la sottodirectory creata in precedenza. Questo modello offre flessibilità nella scelta delle chiavi degli oggetti più adatte all'applicazione, con uguale supporto per directory sparse e dense.

Gestione degli accessi

Nei bucket di directory, tutte le impostazioni Blocco dell'accesso pubblico S3 sono abilitate per impostazione predefinita a livello di bucket. Proprietà dell'oggetto S3 è impostata su Proprietario del bucket applicato e le liste di controllo degli accessi (ACL) sono disabilitate. Queste impostazioni non possono essere modificate.

Per impostazione predefinita, gli utenti non dispongono delle autorizzazioni per i bucket di directory e le operazioni S3 Express One Zone. Per concedere le autorizzazioni di accesso per i bucket di directory, puoi utilizzare IAM per creare utenti, gruppi o ruoli e collegare le autorizzazioni a tali identità. Per ulteriori informazioni, consulta AWS Identity and Access Management (IAM) for S3 Express One Zone.

Utilizzo di bucket di directory

Per ulteriori informazioni sull'utilizzo di bucket di directory, consulta gli argomenti seguenti.

Argomenti

- Regole di denominazione dei bucket di directory
- Creazione di un bucket di directory
- Visualizzazione delle proprietà dei bucket di directory
- Gestione delle policy dei bucket per bucket di directory

- Svuotamento di un bucket di directory
- Eliminazione di un bucket di directory
- Elencare i bucket di directory
- Utilizzo HeadBucket con i bucket di directory

Regole di denominazione dei bucket di directory

Quando si crea un bucket di directory in Amazon S3, si applicano le seguenti regole di denominazione dei bucket. Per le regole di denominazione dei bucket per uso generico, consulta Regole di denominazione dei bucket.

Il nome di un bucket di directory è costituito da un nome di base fornito dall'utente e da un suffisso che contiene l'ID della zona di AWS disponibilità in cui si trova il bucket e. `--x-s3`

```
base-name--azid--x-s3
```

Ad esempio, il seguente nome del bucket di directory contiene l'ID zona di disponibilità `usw2-az1`:

```
bucket-base-name--usw2-az1--x-s3
```

> **Note**
>
> Quando crei un bucket di directory utilizzando la console, viene aggiunto automaticamente un suffisso al nome di base fornito. Questo suffisso include l'ID zona di disponibilità della zona di disponibilità scelta.
> Quando crei un bucket di directory utilizzando un'API, devi fornire il suffisso completo, incluso l'ID della zona di disponibilità, nella richiesta. Per un elenco degli ID delle zone di disponibilità, consulta. Zone di disponibilità e regioni S3 Express One Zone

I nomi dei bucket di directory devono:

- Sii unico all'interno della zona prescelta Regione AWS e della zona di disponibilità.
- Il nome deve avere una lunghezza compresa tra 3 (min) e 63 (max) caratteri, incluso il suffisso.
- Essere costituiti solo da lettere minuscole, numeri e trattini (-).

- Iniziare e finire con una lettera o un numero.
- Deve includere il seguente suffisso:. `--azid--x-s3`

Creazione di un bucket di directory

Per iniziare a utilizzare la classe di archiviazione Amazon S3 Express One Zone, crea una bucket di directory. La classe di archiviazione S3 Express One Zone può essere utilizzata solo con i bucket di directory. La classe di archiviazione S3 Express One Zone supporta casi d'uso a bassa latenza e fornisce un'elaborazione dei dati più rapida all'interno di una singola zona di disponibilità. Se l'applicazione è sensibile alle prestazioni e beneficia di latenze PUT e GET di pochi millisecondi, si consiglia di creare un bucket di directory in modo da poter utilizzare la classe di archiviazione S3 Express One Zone.

Esistono due tipi di bucket Amazon S3: i bucket per uso generico e i bucket di directory. È opportuno scegliere il tipo di bucket più adatto ai requisiti applicativi e di prestazioni. I bucket per uso generico sono il tipo di bucket S3 originale. I bucket generici sono consigliati per la maggior parte dei casi d'uso e dei modelli di accesso e consentono di archiviare oggetti in tutte le classi di archiviazione, ad eccezione di S3 Express One Zone. Per ulteriori informazioni sui bucket per uso generico, consulta Panoramica dei bucket.

I bucket di directory utilizzano la classe di archiviazione S3 Express One Zone, che è progettata per l'utilizzo con carichi di lavoro o applicazioni con prestazioni critiche che richiedono una latenza costante di pochi millisecondi. S3 Express One Zone è la prima classe di archiviazione S3 in cui è possibile selezionare una singola zona di disponibilità con la possibilità di co-ubicare l'archiviazione di oggetti con le risorse di calcolo, che offre la massima velocità di accesso possibile. Quando crei un bucket di directory, puoi facoltativamente specificare una Regione AWS e una zona di disponibilità locale per le tue istanze di calcolo Amazon EC2, Amazon Elastic Kubernetes Service o Amazon Elastic Container Service (Amazon ECS) per ottimizzare le prestazioni.

Con S3 Express One Zone, i dati vengono archiviati in modo ridondante su più dispositivi all'interno di una singola zona di disponibilità. S3 Express One Zone è progettato per una disponibilità del 99,95% all'interno di una singola zona di disponibilità ed è supportato dal Service Level Agreement di Amazon S3. Per ulteriori informazioni, consulta Zona di disponibilità singola

I bucket di directory organizzano i dati gerarchicamente in directory, a differenza della struttura di archiviazione piatta dei bucket generici. Non ci sono limiti di prefissi per i bucket di directory e le singole directory possono essere dimensionate orizzontalmente.

Per ulteriori informazioni sui bucket di directory, consulta Bucks di directory.

Nomi dei bucket di directory

I nomi dei bucket di directory devono seguire questo formato e rispettare le regole di denominazione dei bucket di directory:

```
bucket-base-name--azid--x-s3
```

Ad esempio, il seguente nome del bucket di directory contiene l'ID zona di disponibilità `usw2-az1`:

```
bucket-base-name--usw2-az1--x-s3
```

Per ulteriori informazioni sulle regole di denominazione dei bucket di directory, consulta Regole di denominazione dei bucket di directory.

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nella barra di navigazione nella parte superiore della pagina, scegli il nome del file attualmente visualizzato Regione AWS. Quindi, scegli la regione in cui desideri creare un bucket.

   > **Note**
   >
   > Scegli una regione nelle tue vicinanze per ridurre al minimo la latenza e i costi o essere conforme ai requisiti normativi. Gli oggetti archiviati in una regione non la lasciano mai a meno che non vengano trasferiti esplicitamente in un'altra regione. Per un elenco di Amazon S3 Regioni AWS, consulta gli Servizio AWS endpoint in. Riferimenti generali di Amazon Web Services

3. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
4. Scegliere Create bucket (Crea bucket).

   Viene visualizzata la pagina Create bucket (Crea bucket).
5. In Configurazione generale, visualizza Regione AWS dove verrà creato il bucket.
6. In Tipo di bucket, scegli Directory.

> **Note**
>
> - Se hai scelto una regione che non supporta i bucket di directory, l'opzione Tipo di bucket scompare e il tipo di bucket viene impostato per impostazione predefinita su un bucket generico. Per creare un bucket di directory, devi scegliere una regione supportata. Per un elenco delle regioni che supportano i bucket di directory e la classe di storage Amazon S3 Express One Zone, consulta. the section called “Zone di disponibilità e regioni S3 Express One Zone”
> - Dopo aver creato il bucket, non è possibile modificare il tipo di bucket.

Per Zona di disponibilità, scegli una zona di disponibilità locale nei servizi di calcolo. Per un elenco delle zone di disponibilità che supportano i bucket di directory e la classe di storage S3 Express One Zone, consulta. the section called “Zone di disponibilità e regioni S3 Express One Zone”

> **Note**
>
> La zona di disponibilità non può essere modificata dopo che il bucket è stato creato.

7. In Zona di disponibilità, seleziona la casella di controllo per confermare che, in caso di interruzione della zona di disponibilità, i dati potrebbero non essere disponibili o essere persi.

> **Important**
>
> Sebbene i bucket di directory siano archiviati su più dispositivi all'interno di una singola zona di disponibilità, i bucket di directory non archiviano i dati in modo ridondante tra le zone di disponibilità.

8. Per Nome bucket, immetti il nome del bucket di directory.

   I nomi dei bucket di directory devono:

   - Sii unico all'interno della zona prescelta e della zona di disponibilità Regione AWS .
   - Il nome deve avere una lunghezza compresa tra 3 (min) e 63 (max) caratteri, incluso il suffisso.

- Essere costituiti solo da lettere minuscole, numeri e trattini (-).
- Iniziare e finire con una lettera o un numero.
- Deve includere il seguente suffisso:. `--azid--x-s3`

Un suffisso viene aggiunto automaticamente al nome di base fornito quando si crea un bucket di directory utilizzando la console. Questo suffisso include l'ID zona di disponibilità della zona di disponibilità scelta.

Una volta creato il bucket, non è possibile modificarne il nome. Per ulteriori informazioni sulla denominazione dei bucket, consulta Regole di denominazione dei bucket.

> **Important**
>
> Non includere informazioni riservate, come i numeri di account, nel nome del bucket. Il nome bucket è visibile nell'URL che punta agli oggetti nel bucket.

9. In Object Ownership, l'impostazione imposta dal proprietario del Bucket viene abilitata automaticamente e tutte le liste di controllo degli accessi (ACL) sono disabilitate. Per i bucket di directory, le ACL non possono essere abilitate.

   ACL disabilitate

   - Proprietario del bucket applicato (impostazione predefinita): le ACL sono disabilitate e il proprietario del bucket possiede automaticamente e ha il controllo completo di ogni oggetto nel bucket. Le liste di controllo degli accessi (ACL) non influiscono più sulle autorizzazioni di accesso ai dati nel bucket S3. Il bucket utilizza le policy esclusivamente per definire il controllo degli accessi.

     La maggior parte degli attuali casi d'uso in Amazon S3 non richiede più l'uso delle ACL. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

10. Nelle impostazioni Block Public Access per questo bucket, tutte le impostazioni Block Public Access per il bucket di directory vengono abilitate automaticamente. Queste impostazioni non possono essere modificate per i bucket di directory. Per ulteriori informazioni sul blocco dell'accesso pubblico, consulta Blocco dell'accesso pubblico allo storage Amazon S3.
11. Nelle impostazioni di crittografia lato server, Amazon S3 applica la crittografia lato server con chiavi gestite di Amazon S3 (SSE-S3) come livello base di crittografia per tutti i bucket S3. Tutti

i caricamenti di oggetti nei bucket di directory sono crittografati con SSE-S3. Per i bucket di directory, il tipo di crittografia non può essere modificato. Per ulteriori informazioni su SSE-S3, consulta the section called “Chiavi di crittografia gestite da Amazon S3 (SSE-S3)”.

12. Seleziona Crea bucket.

    Dopo aver creato il bucket, puoi aggiungere file e cartelle al bucket. Per ulteriori informazioni, consulta the section called “Utilizzo di oggetti in un bucket di directory”.

Utilizzo degli SDK AWS

SDK for Go

Questo esempio mostra come creare un bucket di directory utilizzando. 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

Questo esempio mostra come creare un bucket di directory utilizzando. 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

Questo esempio mostra come creare un bucket di directory utilizzando. 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

Questo esempio mostra come creare un bucket di directory utilizzando. 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

Questo esempio mostra come creare un bucket di directory utilizzando. 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

Questo esempio mostra come creare un bucket di directory utilizzando. 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

Questo esempio mostra come creare un bucket di directory utilizzando. 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' }
  }
)
```

Usando il AWS CLI

Questo esempio mostra come creare un bucket di directory utilizzando. AWS CLI Per utilizzare il comando, sostituisci i *segnaposto di input dell'utente* con le tue informazioni.

Quando si crea un bucket di directory, è necessario fornire i dettagli di configurazione e utilizzare la seguente convenzione di denominazione: *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
```

Per ulteriori informazioni, consulta create-bucket in. AWS Command Line Interface

Visualizzazione delle proprietà dei bucket di directory

Puoi visualizzare e configurare le proprietà di un bucket di directory Amazon S3 utilizzando la console Amazon S3. Per ulteriori informazioni, consultare Bucks di directory e Che cos'è S3 Express One Zone?.

Utilizzo della console S3

1. Accedi alla AWS Management Console e apri la console di Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Scegli la scheda Bucket di directory.
4. Nell'elenco Bucket di directory, scegli il nome del bucket per il quale desideri visualizzare le proprietà.
5. Scegliere la scheda Properties (Proprietà).
6. Nella scheda Proprietà, puoi visualizzare le seguenti proprietà del bucket:

- Panoramica del bucket di directory: puoi visualizzare la zona di disponibilitàRegione AWS, l'Amazon Resource Name (ARN) e la data di creazione del bucket.
- Crittografia predefinita: Amazon S3 applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come il livello base di crittografia per tutti i bucket S3. Per i bucket di directory, questa impostazione non può essere modificata. Amazon S3 crittografa di un oggetto prima di salvarlo su disco e lo decritta quando lo scarichi. Per ulteriori informazioni, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3.

  Per ulteriori informazioni sulle funzionalità supportate per i bucket di directory, consulta Funzionalità di S3 Express One Zone.

Gestione delle policy dei bucket per bucket di directory

Puoi aggiungere, eliminare, aggiornare e visualizzare le policy dei bucket per i bucket di directory Amazon S3 utilizzando la console Amazon S3 e gli SDK. AWS Per ulteriori informazioni, consulta i seguenti argomenti. Per ulteriori informazioni sulle azioni supportate AWS Identity and Access Management (IAM) e sui codici di condizione per S3 Express One Zone, consulta. AWS Identity and Access Management (IAM) per S3 Express One Zone Per policy dei bucket di esempio per bucket di directory, consulta Esempi di policy dei bucket di directory per S3 Express One Zone.

Argomenti

- Aggiunta di una policy di bucket
- Visualizzazione di una policy del bucket
- Eliminazione di una policy del bucket

Aggiunta di una policy di bucket

Per aggiungere una policy sui bucket a un bucket di directory, puoi utilizzare la console Amazon S3, gli SDK o AWS il. AWS CLI

Utilizzo della console S3

Per creare o modificare una policy di bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Scegli la scheda Bucket di directory.
4. Nell'elenco Bucket di directory, scegli il nome del bucket in cui desideri caricare le cartelle o i file.
5. Scegli la scheda Autorizzazioni.
6. In Bucket Policy (Policy del bucket) scegliere Edit (Modifica). Viene visualizzata la pagina Edit bucket policy (Modifica policy di bucket).
7. Per generare automaticamente una policy, scegli Policy generator.

   Se scegli Policy generator, AWS Policy Generator si apre in una nuova finestra.

   Se non desideri utilizzare il AWS Policy Generator, puoi aggiungere o modificare le istruzioni JSON nella sezione Policy.

   a. Nella pagina AWS Policy Generator (Generatore di policy AWS), per 'opzione Select Type of Policy (Seleziona tipo di policy), scegli S3 Bucket Policy (Policy di bucket S3).
   b. Aggiungi un'istruzione inserendo le informazioni nei campi forniti, quindi scegli Aggiungi istruzione. Ripeti questo passaggio per tutte le istruzioni che desideri aggiungere. Per ulteriori informazioni su questi campi, consulta Riferimento agli elementi delle policy IAM JSON nella Guida per l'utente IAM.

      > **Note**
      >
      > Per comodità, la pagina Modifica policy del bucket mostra il Bucket ARN (Amazon Resource Name) del bucket corrente sopra il campo di testo Policy. Puoi copiare questo ARN per utilizzarlo nelle istruzioni alla pagina Generatore di policy di AWS .

   c. Dopo aver aggiunto le istruzioni, scegli Genera policy.
   d. Copia il testo della policy generata, scegli Chiudi e torna alla pagina Modifica policy del bucket nella console di Amazon S3.
8. Nella casella Policy, modifica la policy esistente o incolla la bucket policy dal Policy Generator. AWS Assicurati di risolvere gli avvisi di sicurezza, gli errori, gli avvisi generali e i suggerimenti prima di salvare la tua policy.

   > **Note**
   >
   > Le policy di bucket sono limitate a una dimensione di 20 KB.

9. Scegli Save changes (Salva modifiche), che ti riporterà alla pagina Permissions (Autorizzazioni).

Utilizzo degli SDK AWS

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);
        }
    }
```

Usando il AWS CLI

Questo esempio mostra come aggiungere una policy bucket a un bucket di directory utilizzando. AWS CLI Per utilizzare il comando, sostituisci i *segnaposto di input dell'utente* con le tue informazioni.

```
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/"
        }
    ]
}
```

Per ulteriori informazioni, vedere in. put-bucket-policy AWS Command Line Interface

Visualizzazione di una policy del bucket

Per visualizzare una policy relativa ai bucket per un bucket di directory, usa i seguenti esempi.

Usando il AWS CLI

Questo esempio mostra come visualizzare la policy del bucket allegata a un bucket di directory utilizzando. AWS CLI Per utilizzare il comando, sostituisci i *segnaposto di input dell'utente* con le tue informazioni.

```
aws s3api get-bucket-policy --bucket bucket-base-name--azid--x-s3
```

Per ulteriori informazioni, vedere get-bucket-policyin. AWS Command Line Interface

Eliminazione di una policy del bucket

Per eliminare una policy relativa ai bucket per un bucket di directory, usa i seguenti esempi.

Utilizzo degli SDK AWS

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);
      }
```

Usando il AWS CLI

Questo esempio mostra come eliminare una policy bucket per un bucket di directory utilizzando. AWS CLI Per utilizzare il comando, sostituisci i *segnaposto di input dell'utente* con le tue informazioni.

```
aws s3api delete-bucket-policy --bucket bucket-base-name--azid--x-s3
```

Per ulteriori informazioni, vedere delete-bucket-policyin. AWS Command Line Interface

Svuotamento di un bucket di directory

Puoi svuotare un bucket di directory Amazon S3 utilizzando la console Amazon S3. Per ulteriori informazioni sui bucket di directory, consulta Bucks di directory.

Prima di svuotare un bucket di directory, tieni presente quanto segue:

- Quando si svuota un bucket di directory, si eliminano tutti gli oggetti ma si mantiene il bucket di directory.
- Dopo aver svuotato un bucket di directory, l'azione vuota non può essere annullata.
- Gli oggetti che vengono aggiunti al bucket di directory mentre è in corso l'azione del bucket vuoto potrebbero essere eliminati.

Se desideri eliminare anche il bucket, tieni presente quanto segue:

- Tutti gli oggetti nel bucket di directory devono essere eliminati prima di poter eliminare il bucket stesso.
- I caricamenti in più parti in corso nel bucket di directory devono essere interrotti prima di poter eliminare il bucket stesso.

> **Note**
>
> Il `s3 rm` comando tramite AWS Command Line Interface (CLI), l'`delete`operazione tramite Mountpoint e il pulsante di opzione Empty bucket tramite (CLI) non AWS Management Console sono in grado di eliminare i caricamenti multiparte in corso in un bucket di directory. Per eliminare questi caricamenti multiparte in corso, utilizzate l'`ListMultipartUploads`operazione per elencare i caricamenti multiparte in corso nel

bucket e utilizzate l'operazione per interrompere tutti i caricamenti multiparte in corso. `AbortMultupartUpload`

Per eliminare un bucket di directory, consulta Eliminazione di un bucket di directory. Per interrompere un caricamento in più parti in corso, consulta. the section called “Interruzione di un caricamento in più parti”

Per svuotare un bucket per uso generico, consulta Svuotamento di un bucket.

Utilizzo della console S3

Per svuotare un bucket di directory

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Scegli la scheda Bucket di directory.
4. Scegli il pulsante di opzione accanto al nome del bucket che desideri svuotare, quindi scegli Vuoto.
5. Nella pagina Svuota bucket conferma che desideri svuotare il bucket inserendo **`permanently delete`** nel campo di testo e quindi scegli Svuota.
6. Monitora l'avanzamento del processo di svuotamento del secchio nella pagina di stato Secchio vuoto:.

Eliminazione di un bucket di directory

Puoi eliminare solo i bucket di directory Amazon S3 vuoti. Prima di eliminare il bucket di directory, devi eliminare tutti gli oggetti nel bucket e interrompere tutti i caricamenti multiparte in corso.

Per svuotare un bucket di directory, consulta Svuotamento di un bucket di directory. Per interrompere un caricamento multiparte in corso, consulta. the section called “Interruzione di un caricamento in più parti”

Per eliminare un bucket per uso generico, consulta Eliminazione di un bucket.

Utilizzo della console S3

Dopo aver svuotato il bucket della directory e interrotto tutti i caricamenti multiparte in corso, puoi eliminare il bucket.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Scegli la scheda Bucket di directory.
4. Nell'elenco dei bucket di Directory, scegli il pulsante di opzione accanto al bucket che desideri eliminare.
5. Scegli Elimina.
6. Nella pagina Elimina bucket, inserisci il nome del bucket nel campo di testo per confermare l'eliminazione del bucket.

   > **Important**
   >
   > L'eliminazione di un bucket di directory non può essere annullata.

7. Per eliminare il bucket di directory, scegli Elimina bucket.

Utilizzo degli SDK AWS

I seguenti esempi eliminano un bucket di directory utilizzando and. AWS SDK for Java 2.x 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)
```

Utilizzando il AWS CLI

Questo esempio mostra come eliminare un bucket di directory utilizzando. AWS CLI Per utilizzare il comando, sostituisci i *segnaposto di input dell'utente* con le tue informazioni.

```
aws s3api delete-bucket --bucket bucket-base-name--azid--x-s3 --region us-west-2
```

Per ulteriori informazioni, consulta delete-bucket in. AWS Command Line Interface

Elencare i bucket di directory

Gli esempi seguenti mostrano come elencare i bucket di directory utilizzando gli AWS SDK e la AWS CLI.

Utilizzo degli SDK AWS

SDK for Java 2.x

Example

L'esempio seguente elenca i bucket di directory utilizzando. AWS SDK for Java 2.x

```
public static void listBuckets(S3Client s3Client) {
       try {
           ListDirectoryBucketsRequest listDirectoryBucketsRequest =
ListDirectoryBucketsRequest.builder().build();
           ListDirectoryBucketsResponse response =
s3Client.listDirectoryBuckets(listDirectoryBucketsRequest);
           if (response.hasBuckets()) {
               for (Bucket bucket: response.buckets()) {
                   System.out.println(bucket.name());
                   System.out.println(bucket.creationDate());
               }
           }
       }

       catch (S3Exception e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

SDK for Python

Example

L'esempio seguente elenca i bucket di directory utilizzando. AWS SDK for Python (Boto3)

```
import logging
import boto3
from botocore.exceptions import ClientError

def list_directory_buckets(s3_client):
'''
Prints a list of all directory buckets in a Region

:param s3_client: boto3 S3 client
:return: True if there are buckets in the Region, else False
'''
try:
    response = s3_client.list_directory_buckets()
    for bucket in response['Buckets']:
        print (bucket['Name'])
except ClientError as e:
    logging.error(e)
    return False
return True


if __name__ == '__main__':
    region = 'us-east-1'
    s3_client = boto3.client('s3', region_name = region)
    list_directory_buckets(s3_client)
```

AWS SDK for .NET

Example

L'esempio seguente elenca i bucket di directory utilizzando. AWS SDK for .NET

```
var listDirectoryBuckets = await amazonS3Client.ListDirectoryBucketsAsync(new
 ListDirectoryBucketsRequest
{
  MaxDirectoryBuckets = 10
  }).ConfigureAwait(false);
```

SDK for PHP

Example

L'esempio seguente elenca i bucket di directory utilizzando. AWS SDK for PHP

```
require 'vendor/autoload.php';

$s3Client = new S3Client([
    'region'      => 'us-east-1',
]);
$result = $s3Client->listDirectoryBuckets();
```

SDK for Ruby

Example

L'esempio seguente elenca i bucket di directory utilizzando. AWS SDK for Ruby

```
s3 = Aws::S3::Client.new(region:'us-west-1')
s3.list_directory_buckets
```

Utilizzando il AWS CLI

Il comando di `list-directory-buckets` esempio seguente mostra come utilizzare AWS CLI per elencare i bucket di directory nella regione *`us-east-1`*. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
aws s3api list-directory-buckets --region us-east-1
```

Per ulteriori informazioni, consulta la sezione list-directory-buckets nella Documentazione di riferimento della AWS CLI .

Utilizzo **`HeadBucket`** con i bucket di directory

I seguenti esempi AWS SDK mostrano come utilizzare l'operazione `HeadBucket` API per determinare se esiste un bucket di directory Amazon S3 e se si dispone dell'autorizzazione per accedervi.

Utilizzo degli SDK AWS

L' AWS SDK for Java 2.x esempio seguente mostra come determinare se un bucket esiste e se si dispone dell'autorizzazione per accedervi.

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);
   }
}
```

Utilizzando il AWS CLI

Il comando di `head-bucket` esempio seguente mostra come è possibile utilizzare il AWS CLI per determinare se esiste un bucket di directory e se si dispone dell'autorizzazione per accedervi. Per eseguire questo comando, sostituite i segnaposto di input dell'utente con le vostre informazioni.

```
aws s3api head-bucket --bucket bucket-base-name--azid--x-s3
```

Per ulteriori informazioni, consulta la sezione head-bucket nella Documentazione di riferimento della AWS CLI .

Lavorare con oggetti in un bucket di directory

Dopo aver creato un bucket di directory Amazon S3, puoi lavorare con gli oggetti utilizzando la console Amazon S3, AWS Command Line Interface (AWS CLI) e gli SDK. AWS

Per ulteriori informazioni sulle operazioni di massa sugli oggetti con oggetti archiviati nella classe di storage S3 Express One Zone, consulta. Gestione degli oggetti Per ulteriori informazioni sull'importazione, il caricamento, la copia, l'eliminazione e il download di oggetti e la lettura dei metadati dagli oggetti nei bucket di directory, consulta i seguenti argomenti.

Argomenti

Importazione di oggetti in un bucket di directory

Dopo aver creato un bucket di directory in Amazon S3, puoi popolare il nuovo bucket con i dati utilizzando l'azione di importazione. L'importazione è un metodo ottimizzato di creazione di processi Operazioni in batch S3 per copiare oggetti da bucket per uso generico in bucket di directory.

> **Note**
>
> Le seguenti limitazioni si applicano ai processi di importazione:
>
> - Il bucket di origine e il bucket di destinazione si devono trovare nella stessa Regione AWS e nello stesso account.
> - Il bucket di origine non può essere un bucket di directory.
> - Gli oggetti di dimensioni superiori a 5 GB non sono supportati e verranno omessi dall'operazione di copia.
> - Gli oggetti nelle classi di archiviazione Glacier Flexible Retrieval, Glacier Deep Archive, livello di accesso archivio Intelligent-Tiering e livello Intelligent-Tiering Deep Archive devono essere ripristinati prima di poter essere importati.
> - Gli oggetti importati con algoritmi di checksum MD5 vengono convertiti per utilizzare i checksum CRC32.

- Gli oggetti importati utilizzano la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).
- Gli oggetti importati utilizzano la classe di storage Express One Zone, che ha una struttura dei prezzi diversa rispetto alle classi di archiviazione utilizzate dai bucket generici. Considera questa differenza di costo durante l'importazione di un gran numero di oggetti.

Durante la configurazione di un processo di importazione, specifica il bucket o il prefisso di origine da cui verranno copiati gli oggetti esistenti. Inoltre, fornisci un ruolo AWS Identity and Access Management (IAM) che dispone delle autorizzazioni per accedere agli oggetti di origine. Amazon S3 avvia quindi un processo Operazioni in batch che copia gli oggetti e applica automaticamente le impostazioni della classe di archiviazione e del checksum appropriate.

Per configurare i processi di importazione, utilizza la console Amazon S3.

Utilizzo della console Amazon S3

Per importare oggetti in un bucket di directory

1. Accedere alla AWS Management Console e aprire la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Bucket, quindi seleziona la scheda Bucket di directory. Seleziona il pulsante di opzione accanto al bucket di directory in cui desideri importare gli oggetti.
3. Seleziona Importa.
4. Per Origine, inserisci il bucket per uso generico (o il percorso del bucket incluso il prefisso) contenente gli oggetti che desideri importare. Per scegliere un bucket per uso generico esistente da un elenco, seleziona Browse S3.
5. Per Autorizzazione ad accedere e copiare gli oggetti di origine, esegui una delle seguenti operazioni per specificare un ruolo IAM con le autorizzazioni necessarie per importare gli oggetti di origine:
   - Per consentire ad Amazon S3 di creare automaticamente un nuovo ruolo IAM, scegli Crea un nuovo ruolo IAM.

> **Note**
>
> Se gli oggetti di origine sono crittografati con crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS), non selezionare l'opzione Crea un nuovo Ruolo IAM. Specifica invece un ruolo IAM esistente che disponga dell'autorizzazione `kms:Decrypt`.
> Amazon S3 utilizzerà questa autorizzazione per decrittografare gli oggetti. Durante il processo di importazione, Amazon S3 crittograferà nuovamente tali oggetti utilizzando la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).

- Per scegliere un ruolo IAM esistente da un elenco, seleziona Scegli tra ruoli IAM esistenti.
- Per specificare un ruolo IAM esistente inserendo il relativo nome della risorsa Amazon (ARN), seleziona Inserisci ARN ruolo IAM, quindi inserisci l'ARN nel campo corrispondente.

6. Rivedi le informazioni visualizzate nelle sezioni Destinazione e Impostazioni degli oggetti copiati. Se le informazioni nella sezione Destinazione sono corrette, scegli Importa per avviare il processo di copia.

   La console Amazon S3 visualizza lo stato del nuovo processo nella pagina Operazioni in batch. Per ulteriori informazioni sul processo, scegli il pulsante di opzione accanto al nome del processo, quindi nel menu Azioni, scegli Visualizza dettagli. Per aprire il bucket di directory in cui verranno importati gli oggetti, scegli Visualizza la destinazione di importazione.

Utilizzo di Operazioni in batch con S3 Express One Zone

Puoi utilizzare Operazioni in batch Amazon S3 per eseguire operazioni su oggetti archiviati in bucket S3. Per ulteriori informazioni su Operazioni in batch S3, consulta Esecuzione di operazioni in batch su larga scala su oggetti Amazon S3.

Nei seguenti argomenti viene illustrata l'esecuzione di operazioni batch sugli oggetti archiviati nella classe di storage S3 Express One Zone nei bucket di directory.

Argomenti

- Utilizzo di Operazioni in batch con bucket di directory
- Differenze principali

Utilizzo di Operazioni in batch con bucket di directory

È possibile eseguire l'operazione Copy e le operazioni della AWS Lambdafunzione Invoke sugli oggetti archiviati nei bucket di directory. Con Copy, è possibile copiare oggetti tra bucket dello stesso tipo (ad esempio, da un bucket di directory a un bucket di directory). Inoltre, puoi copiare oggetti tra bucket per uso generico e bucket di directory. Con Invoca funzione AWS Lambda, puoi usare una funzione Lambda per eseguire azioni sugli oggetti nel bucket di directory con il codice che definisci.

Copia di oggetti

Puoi copiare tra lo stesso tipo di bucket o tra bucket di directory e bucket per uso generico. Quando copi in un bucket di directory, devi utilizzare il formato Amazon Resource Name (ARN) corretto per questo tipo di bucket. Il formato ARN per un bucket di directory è `arn:aws:s3express:`*`region`*`:`*`account-id`*`:bucket/`*`bucket-base-name`*`--x-s3`.

Puoi anche popolare il bucket di directory con dati utilizzando l'azione Importa nella console S3. L'azione Importa è un metodo ottimizzato di creazione di processi Operazioni in batch S3 per copiare oggetti da bucket per uso generico in bucket di directory. Per i processi di copia Importa da bucket per uso generico a bucket di directory, S3 genera automaticamente un manifesto. Per ulteriori informazioni, consulta Importazione di oggetti in un bucket di directory e Specificazione di un manifesto.

Richiamo di funzioni Lambda () **`LambdaInvoke`**

Esistono requisiti speciali per l'utilizzo di Operazioni in batch per richiamare funzioni Lambda che agiscono su bucket di directory. Ad esempio, è necessario strutturare la richiesta Lambda utilizzando uno schema di invocazione v2 JSON e specificare InvocationSchemaVersion 2.0 quando si crea il lavoro. Per ulteriori informazioni, consulta la funzione Invoke. AWS Lambda

Differenze principali

Di seguito è riportato un elenco di differenze chiave quando si utilizzano le operazioni Batch per eseguire operazioni in blocco su oggetti archiviati in bucket di directory con la classe di storage S3 Express One Zone:

- Amazon S3 esegue automaticamente la crittografa di tutti i nuovi oggetti caricati in un bucket S3. La configurazione di crittografia predefinita di un bucket S3 è sempre abilitata ed è impostata come minimo sulla crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3). Per i bucket di directory, è supportato solo SSSE-S3. Se si effettua una `CopyObject` richiesta che imposta la

crittografia lato server con chiavi fornite dal cliente (SSE-C) o la crittografia lato server con chiavi () (SSE-KMS) su un bucket di AWS KMS directory AWS Key Management Service (origine o destinazione), la risposta restituisce un errore HTTP. `400 (Bad Request)`

- Gli oggetti nei bucket di directory non possono essere taggati. Puoi specificare solo un set di tag vuoto. Per impostazione predefinita, Operazioni in batch copia i tag. Se copi un oggetto con tag tra bucket generici e bucket di directory, ricevi una risposta. `501 (Not Implemented)`
- S3 Express One Zone ti offre la possibilità di scegliere l'algoritmo di checksum utilizzato per convalidare i dati durante i caricamenti o i download. Puoi selezionare uno dei seguenti algoritmi di controllo dell'integrità dei dati Secure Hash Algorithms (SHA) o Cyclic Redundancy Check (CRC): CRC32, CRC32, SHA-1 e SHA-256. I checksum basati su MD5 non sono supportati con la classe di storage S3 Express One Zone.
- Per impostazione predefinita, tutti i bucket Amazon S3 impostano l'impostazione S3 Object Ownership su bucket owner enforced e le liste di controllo degli accessi (ACL) sono disabilitate. Per i bucket di directory, questa impostazione non può essere modificata. È possibile copiare un oggetto da bucket per uso generico in bucket di directory. Tuttavia, non puoi sovrascrivere l'ACL predefinito quando copi da o verso un bucket di directory.
- A prescindere dalla modalità di specifica del manifesto, l'elenco stesso deve essere archiviato in un bucket per uso generico. Le Operazioni Batch non possono importare i manifesti esistenti da (o salvare i manifesti generati in) bucket di directory. Tuttavia, gli oggetti descritti all'interno del manifesto possono essere archiviati in bucket di directory.
- Batch Operations non può specificare un bucket di directory come posizione in un report di S3 Inventory. I report di inventario non supportano i bucket di directory. È possibile creare un file manifesto per gli oggetti all'interno di un bucket di directory utilizzando l'operazione `ListObjectsV2` API per elencare gli oggetti. È quindi possibile inserire l'elenco in un file CSV.

Concessione dell'accesso per

Per eseguire processi di copia, è necessario disporre delle autorizzazioni seguenti:

- Per copiare oggetti da un bucket di directory a un altro, è necessario disporre dell'autorizzazione `s3express:CreateSession`.
- Per copiare oggetti da bucket di directory in bucket per uso generico, è necessario disporre dell'autorizzazione `s3express:CreateSession` e dell'autorizzazione `s3:PutObject` per scrivere la copia dell'oggetto nel bucket di destinazione.

- Per copiare oggetti da bucket generici a bucket di directory, è necessario disporre dell'`s3express:CreateSession`autorizzazione e dell'`s3:GetObject`autorizzazione per leggere l'oggetto di origine che viene copiato.

  Per ulteriori informazioni, consulta CopyObject in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).
- Per richiamare una funzione Lambda, è necessario concedere le autorizzazioni alla risorsa in base alla funzione Lambda. Per determinare quali autorizzazioni sono necessarie, controlla le operazioni API corrispondenti.

Caricamento di un oggetto in un bucket di directory

Dopo aver creato un bucket di directory Amazon S3, puoi caricarvi oggetti. Gli esempi seguenti mostrano come caricare un oggetto in un bucket di directory utilizzando la console S3 e gli SDK. AWS Per informazioni sulle operazioni di caricamento di oggetti in blocco con S3 Express One Zone, consulta. Gestione degli oggetti

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Scegli la scheda Bucket di directory.
4. Scegli il nome del bucket in cui vuoi caricare le cartelle o i file.
5. Nell'elenco Oggetti, scegli Carica.
6. Nella pagina Carica, effettuate una delle seguenti operazioni:
   - Trascina i file e le cartelle nell'area di caricamento punteggiata.
   - Scegli Aggiungi file o Aggiungi cartella, scegli i file o le cartelle da caricare, quindi scegli Apri o Carica.
7. In Checksum, scegli la funzione Checksum che desideri utilizzare.

   (Facoltativo) Se state caricando un singolo oggetto di dimensioni inferiori a 16 MB, potete anche specificare un valore di checksum precalcolato. Quando fornisci un valore precalcolato, Amazon S3 lo confronta con il valore calcolato utilizzando la funzione di checksum selezionata. Se i valori non corrispondono, il caricamento non inizierà.

8. Le opzioni nelle sezioni Autorizzazioni e Proprietà vengono impostate automaticamente sulle impostazioni predefinite e non possono essere modificate. Block Public Access è abilitato automaticamente e S3 Versioning e S3 Object Lock non possono essere abilitati per i bucket di directory.

   (Facoltativo) Se desideri aggiungere metadati in coppie chiave-valore ai tuoi oggetti, espandi la sezione Proprietà, quindi nella sezione Metadati scegli Aggiungi metadati.

9. Per caricare i file e le cartelle elencati, scegliete Carica.

   Amazon S3 caricherà i tuoi oggetti e le tue cartelle. Al termine del caricamento viene visualizzato un messaggio di esito positivo nella pagina Carica: stato.

Utilizzo degli AWS SDK

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()
```

Utilizzando il AWS CLI

Il seguente comando di `put-object` esempio mostra come utilizzare AWS CLI per caricare un oggetto da Amazon S3. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
aws s3api put-object --bucket bucket-base-name--azid--x-s3 --key sampleinut/file001.bin
 --body bucket-seed/file001.bin
```

Per ulteriori informazioni, consulta la sezione put-object nella Documentazione di riferimento della AWS CLI .

Utilizzo di caricamenti multiparte con bucket di directory

Puoi utilizzare il processo di caricamento in più parti per caricare un singolo oggetto come set di parti. Ciascuna parte è una parte contigua dei dati dell'oggetto. È possibile caricare queste parti dell'oggetto in modo indipendente e in qualsiasi ordine. Se la trasmissione di una parte non riesce, è possibile ritrasmettere tale parte senza influire sulle altre. Una volta caricate tutte le parti dell'oggetto, Amazon S3 le assembla e crea l'oggetto. In generale, quando la dimensione dell'oggetto raggiunge i 100 MB, si consiglia di valutare la possibilità di eseguire caricamenti in più parti anziché caricare l'oggetto in una singola operazione.

Il caricamento in più parti comporta i vantaggi riportati di seguito.

- Velocità effettiva migliorata: puoi caricare le parti in parallelo per migliorare la velocità effettiva.
- Ripristino rapido da qualsiasi problema di rete: le dimensioni delle parti più piccole riducono al minimo l'impatto del riavvio di un caricamento non riuscito a causa di un errore di rete.
- Messa in pausa e ripresa dei caricamenti dell'oggetto: puoi caricare le parti dell'oggetto nel corso del tempo. Dopo aver avviato un caricamento in più parti, non esiste una data di scadenza. È necessario completare o interrompere in modo esplicito il caricamento in più parti.
- Avvio di un caricamento prima di conoscere la dimensione finale dell'oggetto: puoi caricare un oggetto mentre viene creato.

Ti consigliamo di utilizzare i caricamenti in più parti nei seguenti modi:

- Se carichi oggetti di grandi dimensioni su una rete stabile a larghezza di banda elevata, utilizza caricamenti in più parti per massimizzare l'uso della larghezza di banda disponibile caricando parti dell'oggetto in parallelo per prestazioni multithread.

- Se stai caricando su una rete irregolare, utilizza caricamenti in più parti per aumentare la resilienza agli errori di rete evitando il riavvio del caricamento. Quando utilizzi caricamenti in più parti, devi riprovare a caricare solo le parti che sono state interrotte durante il caricamento. Non è necessario riavviare il caricamento dell'oggetto dall'inizio.

Quando utilizzi caricamenti multiparte per caricare oggetti nella classe di storage Amazon S3 Express One Zone nei bucket di directory, il processo di caricamento multiparte è simile al processo di utilizzo del caricamento multiparte per caricare oggetti in bucket generici. Tuttavia, non vi sono alcune differenze importanti.

Per ulteriori informazioni sull'utilizzo di caricamenti in più parti per caricare oggetti su S3 Express One Zone, consulta i seguenti argomenti.

Argomenti

Il processo di caricamento in più parti

Un caricamento in più parti è un processo in tre fasi:

- Avviate il caricamento.
- Le parti dell'oggetto vengono caricate.
- Dopo aver caricato tutte le parti, si completa il caricamento multiparte.

Dopo aver ricevuto la richiesta di caricamento multiparte completa, Amazon S3 costruisce l'oggetto a partire dalle parti caricate e puoi quindi accedere all'oggetto come faresti con qualsiasi altro oggetto nel tuo bucket.

Avvio del caricamento in più parti

Quando invii una richiesta di avvio di un caricamento in più parti, Amazon S3 restituisce una risposta con un ID di caricamento, che è un identificativo univoco per il caricamento in più parti. È necessario includere questo ID di caricamento ogni volta che si caricano o si elencano le parti oppure ogni volta che si completa o si interrompe un caricamento.

Caricamento delle parti

Quando si carica una parte, oltre all'ID di caricamento è necessario specificare il numero della parte. Quando utilizzi un caricamento in più parti con S3 Express One Zone, i codici articolo multiparte devono essere numeri di parte consecutivi. Se tenti di completare una richiesta di caricamento in più parti con numeri di parte non consecutivi, viene generato un errore `HTTP 400 Bad Request` (Ordine delle parti non valido).

Un numero di parte identifica in modo univoco una parte e la sua posizione nell'oggetto che state caricando. Se caricate una nuova parte utilizzando lo stesso numero di parte di una parte caricata in precedenza, la parte caricata in precedenza viene sovrascritta.

Ogni volta che carichi una parte, Amazon S3 restituisce un'intestazione tag di entità (ETag) nella risposta corrispondente. Per ogni caricamento di parte è necessario registrare il numero della parte e il valore ETag. I valori ETag per tutti i caricamenti di parti di oggetti rimarranno gli stessi, ma a ciascuna parte verrà assegnato un numero di parte diverso. Occorre includere questi valori nella successiva richiesta di complemento del caricamento in più parti.

Amazon S3 esegue automaticamente la crittografa di tutti i nuovi oggetti caricati in un bucket S3. Quando si esegue un caricamento in più parti, se non si specificano le informazioni di crittografia nella richiesta, l'impostazione di crittografia delle parti caricate viene impostata sulla configurazione di crittografia predefinita del bucket di destinazione. La configurazione di crittografia predefinita di un bucket Amazon S3 è sempre abilitata ed è impostata come minimo sulla crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3). Per i bucket di directory, è supportato solo SSE-S3. Per ulteriori informazioni, consulta Crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).

Completamento del caricamento in più parti

Quando completi un caricamento in più parti, Amazon S3 crea l'oggetto concatenando le parti in ordine crescente in base al numero di parte. Una volta completata la richiesta, le parti non esisteranno più.

La tua richiesta di caricamento multiparte completa deve includere l'ID di caricamento e un elenco di entrambi i numeri di parte e i valori ETag corrispondenti. La risposta di Amazon S3 include un ETag

che identifica in modo univoco i dati oggetto combinati. Questo ETag non è un hash MD5 dei dati dell'oggetto.

Elenchi dei caricamenti in più parti

È possibile elencare le parti di un caricamento in più parti specifico o tutti i caricamenti in più parti in corso. L'operazione di creazione dell'elenco delle parti restituisce informazioni sulle parti coinvolte in un caricamento in più parti specifico. Per ogni richiesta di elenco delle parti, Amazon S3 restituisce informazioni sulle parti per il caricamento in più parti specificato, fino a un massimo di 1000 parti. Se nel caricamento in più parti sono presenti più di 1000 parti, è necessario utilizzare la paginazione per recuperare tutte le parti.

L'elenco di parti restituito non include le parti il cui caricamento non è stato completato. L'operazione list multipart uploads (elenco dei caricamenti in più parti) consente di ottenere l'elenco dei caricamenti in più parti in corso.

Un caricamento in più parti in corso è un caricamento avviato, ma non ancora completato o annullato. Ogni richiesta restituisce al massimo 1.000 caricamenti in più parti. Se sono in corso più di 1.000 caricamenti in più parti, è necessario inviare richieste aggiuntive per recuperare i caricamenti rimanenti. Utilizza l'elenco restituito solo per la verifica. Non utilizzarlo per inviare la richiesta complete multipart upload (completamento del caricamento in più parti). Al contrario, mantieni il tuo elenco dei numeri delle parti specificato durante il caricamento delle parti e i valori ETag corrispondenti restituiti da Amazon S3.

Per ulteriori informazioni sulle inserzioni con caricamento multiparte, consulta il riferimento ListPartsall'API di Amazon Simple Storage Service.

Checksum con operazioni di caricamento in più parti

Quando si carica un oggetto, è possibile specificare un algoritmo di checksum per verificare l'integrità dell'oggetto. MD5 non è supportato per i bucket di directory. Puoi specificare uno dei seguenti algoritmi di controllo dell'integrità dei dati tramite Secure Hash Algorithms (SHA) o Cyclic Redundancy Check (CRC):

- CRC32
- CRC32C
- SHA-1
- SHA-256

Puoi utilizzare l'API REST di Amazon S3 o gli AWS SDK per recuperare il valore del checksum per le singole parti utilizzando o. `GetObject` `HeadObject` Se desideri recuperare i valori di checksum per singole parti di caricamenti in più parti ancora in corso, puoi utilizzare `ListParts`.

> **Important**
>
> Quando si utilizzano gli algoritmi di checksum precedenti, i numeri di parte multiparte devono utilizzare numeri di parte consecutivi. Se tenti di completare una richiesta di caricamento in più parti con numeri di parte non consecutivi, Amazon S3 genera `HTTP 400 Bad Request` un errore (Invalid Part Order).

Per ulteriori informazioni sul funzionamento dei checksum con oggetti in più parti, consulta Verifica dell'integrità degli oggetti.

Operazioni simultanee di caricamento in più parti

In un ambiente di sviluppo distribuito, l'applicazione può avviare diversi aggiornamenti sullo stesso oggetto contemporaneamente. Ad esempio, l'applicazione potrebbe avviare diversi caricamenti in più parti utilizzando la stessa chiave oggetto. Per ciascuno di questi caricamenti, l'applicazione può quindi caricare le parti e inviare una richiesta di completamento del caricamento ad Amazon S3 per creare l'oggetto. Per S3 Express One Zone, l'ora di creazione dell'oggetto è la data di completamento del caricamento in più parti.

> **Important**
>
> Il controllo delle versioni non è supportato per gli oggetti archiviati nei bucket di directory.

Caricamenti e prezzi in più parti

Una volta avviato un caricamento in più parti, Amazon S3 mantiene tutte le parti finché il caricamento non viene completato o interrotto. Per tutta la durata del processo, all'utente vengono fatturati i costi per lo storage, la larghezza di banda e le richieste per questo tipo di caricamento e per le parti associate. Se interrompi il caricamento in più parti, Amazon S3 elimina gli elementi del caricamento e tutte le parti che hai caricato e non ti viene più addebitato alcun costo. Non sono previsti costi di eliminazione anticipata per l'eliminazione di caricamenti multiparte incompleti, indipendentemente dalla classe di archiviazione specificata. Per ulteriori informazioni sui prezzi, consulta la sezione Prezzi di Amazon S3.

> **Important**
>
> Se la richiesta di caricamento multiparte completa non viene inviata correttamente, le parti dell'oggetto non vengono assemblate e un oggetto non viene creato. Ti viene addebitato tutto lo spazio di storage associato alle parti caricate. È importante completare il caricamento in più parti per creare l'oggetto o interrompere il caricamento in più parti per rimuovere le parti caricate.
> Prima di poter eliminare un bucket di directory, devi completare o interrompere tutti i caricamenti multiparte in corso. I bucket di directory non supportano le configurazioni S3 Lifecycle. Se necessario, puoi elencare i caricamenti multiparte attivi, quindi interrompere i caricamenti e quindi eliminare il bucket.

Operazioni e autorizzazioni dell'API di caricamento in più parti

Per consentire l'accesso alle operazioni dell'API di gestione degli oggetti su un bucket di directory, concedi l'`s3express:CreateSession`autorizzazione in una policy di bucket o in una policy basata sull'identità AWS Identity and Access Management (IAM).

Per eseguire le operazioni di caricamento in più parti, devi disporre delle autorizzazioni necessarie. Puoi utilizzare le bucket policy o le policy basate sull'identità IAM per concedere ai principali IAM le autorizzazioni per eseguire queste operazioni. Nella tabella riportata di seguito sono elencate le autorizzazioni richieste per le varie operazioni di caricamento in più parti.

È possibile identificare l'iniziatore di un caricamento in più parti tramite l'elemento. `Initiator` Se l'iniziatore è un Account AWS, questo elemento fornisce le stesse informazioni dell'elemento. `Owner` Se è un utente IAM, questo elemento fornisce l'ARN e il nome visualizzato dell'utente.

Azione	Autorizzazioni richieste
Creazione di un caricamento in più parti	Per creare il caricamento in più parti, devi avere il permesso di eseguire l'`s3express:CreateSession` azione sul bucket di directory.
Avvia un caricamento in più parti	Per avviare il caricamento in più parti, devi avere il permesso di eseguire l'`s3express:CreateSession` azione sul bucket di directory.

Azione	Autorizzazioni richieste
Carica una parte	Per caricare una parte, devi avere il permesso di eseguire l'`s3express:CreateSession` azione nel bucket di directory. Affinché l'iniziatore carichi una parte, il proprietario del bucket deve consentire all'iniziatore di eseguire l'`s3express:CreateSession` azione sul bucket di directory.
Carica una parte (copia)	Per caricare una parte, devi avere il permesso di eseguire l'`s3express:CreateSession` azione sul bucket di directory. Perché l'iniziatore possa caricare una parte di un oggetto, il proprietario del bucket deve consentire all'iniziatore di eseguire l'operazione `s3express:CreateSession` sull'oggetto.
Completamento del caricamento in più parti	Per completare un caricamento in più parti, è necessario essere autorizzati a eseguire l'`s3express:CreateSession` azione sul bucket di directory. Affinché l'iniziatore completi un caricamento in più parti, il proprietario del bucket deve consentire all'iniziatore di eseguire l'azione sull'oggetto. `s3express:CreateSession`
Interrompere un caricamento in più parti	Per interrompere un caricamento in più parti, devi avere il permesso di eseguire l'azione. `s3express:CreateSession` Affinché l'iniziatore possa interrompere un caricamento in più parti, all'iniziatore deve essere concesso l'autorizzazione esplicita all'accesso per eseguire l'azione. `s3express:CreateSession`
Elenca le parti	Per elencare le parti in un caricamento in più parti, devi avere il permesso di eseguire l'`s3express:CreateSession` azione sul bucket di directory.
Elenco di caricamenti in più parti in corso	Per elencare i caricamenti multiparte in corso in un bucket, devi avere il permesso di eseguire l'azione su quel bucket. `s3:ListBucketMultipartUploads`

Supporto operativo tramite API per caricamenti in più parti

Le seguenti sezioni del riferimento all'API di Amazon Simple Storage Service descrivono le operazioni dell'API REST di Amazon S3 per caricamenti multiparte.

- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- AbortMultipartUpload
- ListParts
- ListMultipartUploads

Esempi

Per utilizzare un caricamento in più parti per caricare un oggetto su S3 Express One Zone in un bucket di directory, consulta gli esempi seguenti.

Argomenti

- Creazione di un caricamento in più parti
- Caricamento delle parti di un caricamento in più parti
- Completamento di un caricamento in più parti
- Interruzione di un caricamento in più parti
- Creazione di un'operazione di copia di caricamento in più parti
- Elenco dei caricamenti multiparte in corso
- Elencare le parti di un caricamento in più parti

Creazione di un caricamento in più parti

Gli esempi seguenti mostrano come creare un caricamento in più parti.

Utilizzo degli SDK 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
```

Usando il AWS CLI

Questo esempio mostra come creare un caricamento multiparte in un bucket di directory utilizzando. AWS CLI*Questo comando avvia un caricamento in più parti nel bucket di directory bucket-base-name-- azid --x-s3 per l'oggetto KEY_NAME. Per utilizzare il comando, sostituisci i segnaposto di input dell'utente con le tue informazioni.*

```
aws s3api create-multipart-upload --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
```

Per ulteriori informazioni, vedere create-multipart-uploadin. AWS Command Line Interface

Caricamento delle parti di un caricamento in più parti

Gli esempi seguenti mostrano come caricare parti di un caricamento in più parti.

Utilizzo degli SDK AWS

SDK for Java 2.x

L'esempio seguente mostra come suddividere un singolo oggetto in parti e quindi caricare tali parti in un bucket di directory utilizzando l'SDK for Java 2.x.

Example

```
/**
```

```
* This method creates part requests and uploads individual parts to S3 and then
returns all the completed parts
*
* @param s3
* @param bucketName
* @param key
* @param uploadId
* @throws IOException
*/
private static ListCompletedPartmultipartUpload(S3Client s3, String bucketName,
String key, String uploadId, String filePath) throws IOException {

        int partNumber = 1;
        ListCompletedPart completedParts = new ArrayList<>();
        ByteBuffer bb = ByteBuffer.allocate(1024 * 1024 * 5); // 5 MB byte buffer

        // read the local file, breakdown into chunks and process
        try (RandomAccessFile file = new RandomAccessFile(filePath, "r")) {
            long fileSize = file.length();
            int position = 0;
            while (position < fileSize) {
                file.seek(position);
                int read = file.getChannel().read(bb);

                bb.flip(); // Swap position and limit before reading from the
buffer.
                UploadPartRequest uploadPartRequest = UploadPartRequest.builder()
                        .bucket(bucketName)
                        .key(key)
                        .uploadId(uploadId)
                        .partNumber(partNumber)
                        .build();

                UploadPartResponse partResponse = s3.uploadPart(
                        uploadPartRequest,
                        RequestBody.fromByteBuffer(bb));

                CompletedPart part = CompletedPart.builder()
                        .partNumber(partNumber)
                        .eTag(partResponse.eTag())
                        .build();
                completedParts.add(part);

                bb.clear();
```

```
                position += read;
                partNumber++;
            }
        }

        catch (IOException e) {
            throw e;
        }
        return completedParts;
    }
```

SDK for Python

L'esempio seguente mostra come suddividere un singolo oggetto in parti e quindi caricare tali parti in un bucket di directory utilizzando l'SDK per 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
```

Usando il AWS CLI

Questo esempio mostra come suddividere un singolo oggetto in parti e quindi caricare tali parti in un bucket di directory utilizzando. AWS CLI Per utilizzare il comando, sostituisci i *segnaposto di input dell'utente* con le tue informazioni.

```
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
```

Per ulteriori informazioni, consulta upload-part in. AWS Command Line Interface

Completamento di un caricamento in più parti

Gli esempi seguenti mostrano come completare un caricamento in più parti.

Utilizzo degli SDK AWS

SDK for Java 2.x

Gli esempi seguenti mostrano come completare un caricamento in più parti utilizzando l'SDK for Java 2.x.

Example

```
/**
 * This method completes the multipart upload request by collating all the upload
 parts
 * @param s3
```

```
* @param bucketName - for example, 'doc-example-bucket--usw2-az1--x-s3'
* @param key
* @param uploadId
* @param uploadParts
*/
private static void completeMultipartUpload(S3Client s3, String bucketName, String
key, String uploadId, ListCompletedPart uploadParts) {
       CompletedMultipartUpload completedMultipartUpload =
CompletedMultipartUpload.builder()
               .parts(uploadParts)
               .build();

       CompleteMultipartUploadRequest completeMultipartUploadRequest =
               CompleteMultipartUploadRequest.builder()
                       .bucket(bucketName)
                       .key(key)
                       .uploadId(uploadId)
                       .multipartUpload(completedMultipartUpload)
                       .build();

       s3.completeMultipartUpload(completeMultipartUploadRequest);
   }

   public static void multipartUploadTest(S3Client s3, String bucketName, String
key, String localFilePath)  {
       System.out.println("Starting multipart upload for: " + key);
       try {
           String uploadId = createMultipartUpload(s3, bucketName, key);
           System.out.println(uploadId);
           ListCompletedPart parts = multipartUpload(s3, bucketName, key, uploadId,
localFilePath);
           completeMultipartUpload(s3, bucketName, key, uploadId, parts);
           System.out.println("Multipart upload completed for: " + key);
       }

       catch (Exception e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
   }
```

SDK for Python

Gli esempi seguenti mostrano come completare un caricamento in più parti utilizzando l'SDK per 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}')
```

Usando il AWS CLI

Questo esempio mostra come completare un caricamento in più parti per un bucket di directory utilizzando. AWS CLI Per utilizzare il comando, sostituisci i *segnaposto di input dell'utente* con le tue informazioni.

```
aws s3api complete-multipart-upload --bucket bucket-base-name--azid--x-s3 --
key KEY_NAME --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
 --multipart-upload file://parts.json
```

Questo esempio utilizza una struttura JSON che descrive le parti del caricamento in più parti che devono essere riassemblate nel file completo. In questo esempio, il `file://` prefisso viene utilizzato per caricare la struttura JSON da un file nella cartella locale denominata. `parts`

parts.json:

```
parts.json
{
  "Parts": [
    {
      "ETag": "6b78c4a64dd641a58dac8d9258b88147",
      "PartNumber": 1
    }
  ]
}
```

Per ulteriori informazioni, vedere complete-multipart-uploadin. AWS Command Line Interface

Interruzione di un caricamento in più parti

Gli esempi seguenti mostrano come interrompere un caricamento in più parti.

Utilizzo degli SDK AWS

SDK for Java 2.x

L'esempio seguente mostra come interrompere un caricamento in più parti utilizzando l'SDK for Java 2.x.

Example

```
public static void abortMultiPartUploads( S3Client s3, String bucketName ) {

        try {
            ListMultipartUploadsRequest listMultipartUploadsRequest =
 ListMultipartUploadsRequest.builder()
                    .bucket(bucketName)
                    .build();

            ListMultipartUploadsResponse response =
 s3.listMultipartUploads(listMultipartUploadsRequest);
            ListMultipartUpload uploads = response.uploads();

            AbortMultipartUploadRequest abortMultipartUploadRequest;
            for (MultipartUpload upload: uploads) {
                abortMultipartUploadRequest = AbortMultipartUploadRequest.builder()
                        .bucket(bucketName)
                        .key(upload.key())
                        .uploadId(upload.uploadId())
                        .build();

                s3.abortMultipartUpload(abortMultipartUploadRequest);
            }

        }

        catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

SDK for Python

L'esempio seguente mostra come interrompere un caricamento in più parti utilizzando l'SDK per 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')
```

Usando il AWS CLI

L'esempio seguente mostra come interrompere un caricamento in più parti utilizzando. AWS CLI Per utilizzare il comando, sostituisci i *segnaposto di input dell'utente* con le tue informazioni.

```
aws s3api abort-multipart-upload --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
 --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
MAQAAAAB0OxUFeA7LTbWWFS8WYwhrxDxTIDN-pdEEq_agIHqsbg"
```

Per ulteriori informazioni, vedere abort-multipart-uploadin. AWS Command Line Interface

Creazione di un'operazione di copia di caricamento in più parti

Gli esempi seguenti mostrano come copiare oggetti da un bucket a un altro utilizzando un caricamento in più parti.

Utilizzo degli SDK AWS

SDK for Java 2.x

L'esempio seguente mostra come utilizzare un caricamento in più parti per copiare a livello di codice un oggetto da un bucket a un altro utilizzando l'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
 * @param key
 * @return
 */
 private static String createMultipartUpload(S3Client s3, String bucketName, String
 key) {
        CreateMultipartUploadRequest createMultipartUploadRequest =
 CreateMultipartUploadRequest.builder()
                .bucket(bucketName)
                .key(key)
                .build();
```

```
        String uploadId = null;
        try {
            CreateMultipartUploadResponse response =
s3.createMultipartUpload(createMultipartUploadRequest);
            uploadId = response.uploadId();
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return uploadId;
 }

 /**
  * Creates copy parts based on source object size and copies over individual parts
  *
  * @param s3
  * @param sourceBucket
  * @param sourceKey
  * @param destnBucket
  * @param destnKey
  * @param uploadId
  * @return
  * @throws IOException
  */
   public static ListCompletedPart multipartUploadCopy(S3Client s3, String
sourceBucket, String sourceKey, String destnBucket, String destnKey, String
uploadId) throws IOException {

        // Get the object size to track the end of the copy operation.
        HeadObjectRequest headObjectRequest = HeadObjectRequest
                .builder()
                .bucket(sourceBucket)
                .key(sourceKey)
                .build();
        HeadObjectResponse response = s3.headObject(headObjectRequest);
        Long objectSize = response.contentLength();

        System.out.println("Source Object size: " + objectSize);

        // Copy the object using 20 MB parts.
        long partSize = 20 * 1024 * 1024;
        long bytePosition = 0;
        int partNum = 1;
        ListCompletedPart completedParts = new ArrayList<>();
```

```
        while (bytePosition < objectSize) {
            // The last part might be smaller than partSize, so check to make sure
            // that lastByte isn't beyond the end of the object.
            long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);

            System.out.println("part no: " + partNum + ", bytePosition: " +
bytePosition + ", lastByte: " + lastByte);

            // Copy this part.
            UploadPartCopyRequest req = UploadPartCopyRequest.builder()
                    .uploadId(uploadId)
                    .sourceBucket(sourceBucket)
                    .sourceKey(sourceKey)
                    .destinationBucket(destnBucket)
                    .destinationKey(destnKey)
                    .copySourceRange("bytes="+bytePosition+"-"+lastByte)
                    .partNumber(partNum)
                    .build();
            UploadPartCopyResponse res = s3.uploadPartCopy(req);
            CompletedPart part = CompletedPart.builder()
                    .partNumber(partNum)
                    .eTag(res.copyPartResult().eTag())
                    .build();
            completedParts.add(part);
            partNum++;
            bytePosition += partSize;
        }
        return completedParts;
    }


    public static void multipartCopyUploadTest(S3Client s3, String srcBucket, String
srcKey, String destnBucket, String destnKey)  {
        System.out.println("Starting multipart copy for: " + srcKey);
        try {
            String uploadId = createMultipartUpload(s3, destnBucket, destnKey);
            System.out.println(uploadId);
            ListCompletedPart parts = multipartUploadCopy(s3, srcBucket,
srcKey,destnBucket,  destnKey, uploadId);
            completeMultipartUpload(s3, destnBucket, destnKey, uploadId, parts);
            System.out.println("Multipart copy completed for: " + srcKey);
        } catch (Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
```

```
        }
    }
```

SDK for Python

L'esempio seguente mostra come utilizzare un caricamento in più parti per copiare a livello di codice un oggetto da un bucket all'altro utilizzando l'SDK per 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}')
```

Usando il AWS CLI

L'esempio seguente mostra come utilizzare un caricamento in più parti per copiare a livello di codice un oggetto da un bucket a un bucket di directory utilizzando. AWS CLI Per utilizzare il comando, sostituisci i *segnaposto di input dell'utente* con le tue informazioni.

```
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
```

Per ulteriori informazioni, vedere upload-part-copyin. AWS Command Line Interface

Elenco dei caricamenti multiparte in corso

Per elencare i caricamenti multiparte in corso in un bucket di directory, puoi utilizzare gli SDK o il. AWS AWS CLI

Utilizzo degli SDK AWS

SDK for Java 2.x

Gli esempi seguenti mostrano come elencare caricamenti multiparte in corso (incompleti) utilizzando l'SDK for 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

Gli esempi seguenti mostrano come elencare caricamenti multiparte in corso (incompleti) utilizzando l'SDK per 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}')
```

Usando il AWS CLI

Gli esempi seguenti mostrano come elencare i caricamenti multiparte in corso (incompleti) utilizzando. AWS CLI Per utilizzare il comando, sostituisci i *segnaposto di input dell'utente* con le tue informazioni.

```
aws s3api list-multipart-uploads --bucket bucket-base-name--azid--x-s3
```

Per ulteriori informazioni, vedere list-multipart-uploadsin. AWS Command Line Interface

Elencare le parti di un caricamento in più parti

Gli esempi seguenti mostrano come elencare le parti di un caricamento in più parti in un bucket di directory.

Utilizzo degli SDK AWS

SDK for Java 2.x

Gli esempi seguenti mostrano come elencare le parti di un caricamento multiparte in un bucket di directory utilizzando SDK for Java 2.x.

```
public static void listMultiPartUploadsParts( S3Client s3, String bucketName, String
 objKey, String uploadID) {
```

```
        try {
            ListPartsRequest listPartsRequest = ListPartsRequest.builder()
                .bucket(bucketName)
                .uploadId(uploadID)
                .key(objKey)
                .build();

            ListPartsResponse response = s3.listParts(listPartsRequest);
            ListPart parts = response.parts();
            for (Part part: parts) {
                System.out.println("Upload in progress: Part number = \"" +
 part.partNumber() + "\", etag = " + part.eTag());
            }

        }

        catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }


    }
```

SDK for Python

Gli esempi seguenti mostrano come elencare le parti di un caricamento multiparte in un bucket di directory utilizzando SDK per 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')
```

Usando il AWS CLI

Gli esempi seguenti mostrano come elencare le parti di un caricamento in più parti in un bucket di directory utilizzando. AWS CLI Per utilizzare il comando, sostituisci i *segnaposto di input dell'utente* con le tue informazioni.

```
aws s3api list-parts --bucket bucket-base-name--azid--x-s3 --key KEY_NAME --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
```

Per ulteriori informazioni, vedere list-parts in. AWS Command Line Interface

Copia di un oggetto in un bucket di directory

L'operazione copy crea una copia di un oggetto già archiviato in Amazon S3. Puoi copiare oggetti tra bucket di directory e bucket per uso generico. Inoltre, puoi copiare oggetti all'interno di un bucket e tra bucket dello stesso tipo, ad esempio, da un bucket di directory all'altro.

Puoi creare una copia di un oggetto fino a 5 GB in una singola operazione atomica. Tuttavia, per copiare un oggetto di dimensioni superiori a 5 GB, è necessario utilizzare le operazioni API di caricamento multiparte. Per ulteriori informazioni, consulta Utilizzo di caricamenti multiparte con bucket di directory.

Autorizzazioni

Per copiare oggetti, è necessario disporre delle seguenti autorizzazioni:

- Per copiare oggetti da un bucket di directory a un altro, è necessario disporre dell'autorizzazione `s3express:CreateSession`.
- Per copiare oggetti da bucket di directory in bucket per uso generico, è necessario disporre dell'autorizzazione `s3express:CreateSession` e dell'autorizzazione `s3:PutObject` per scrivere la copia dell'oggetto nel bucket di destinazione.
- Per copiare oggetti da bucket generici a bucket di directory, è necessario disporre dell'`s3express:CreateSession`autorizzazione e dell'`s3:GetObject`autorizzazione per leggere l'oggetto di origine che viene copiato.

Per ulteriori informazioni, consulta CopyObject in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Crittografia

Amazon S3 esegue automaticamente la crittografa di tutti i nuovi oggetti caricati in un bucket S3. La configurazione di crittografia predefinita di un bucket S3 è sempre abilitata ed è impostata come minimo sulla crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).

Per i bucket di directory, è supportato solo SSE-S3. Per i bucket generici, è possibile utilizzare SSE-S3 (impostazione predefinita), la crittografia lato server con () chiavi (SSE-KMS), la crittografia lato server a due livelli con AWS Key Management Service chiavi (DSSE-KMS) o la crittografia lato server con chiavi fornite dal cliente (SSE-C).AWS KMS AWS KMS

Se si effettua una richiesta di copia che imposta i parametri SSE-C, SSE-KMS o DSSE-KMS su un bucket di directory come origine o destinazione, la risposta restituisce un errore,

Tag

I bucket di directory non supportano i tag. Se si copia un oggetto con tag da un bucket generico a un bucket di directory, si riceve una risposta HTTP. `501 (Not Implemented)` Per ulteriori informazioni, consulta CopyObject in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

ETag

I tag di entità (ETag) per S3 Express One Zone sono stringhe alfanumeriche casuali e non sono checksum MD5. Per garantire l'integrità degli oggetti, utilizza checksum aggiuntivi.

Checksum aggiuntivi

S3 Express One Zone offre la possibilità di scegliere l'algoritmo di checksum utilizzato per convalidare i dati durante il caricamento o il download. Puoi selezionare uno dei seguenti algoritmi di controllo dell'integrità dei dati Secure Hash Algorithms (SHA) o Cyclic Redundancy Check (CRC): CRC32, CRC32C, SHA-1 e SHA-256. I checksum basati su MD5 non sono supportati con la classe di storage S3 Express One Zone.

Per ulteriori informazioni, consulta Best practice per il checksum S3 aggiuntivo.

Funzionalità supportate

Per ulteriori informazioni su quali funzionalità di Amazon S3 sono supportate per S3 Express One Zone, consulta. In cosa differisce S3 Express One Zone?

Utilizzo della console S3 (copia in un bucket di directory)

Per copiare un oggetto da un bucket generico o da un bucket di directory a un bucket di directory

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Scegli il bucket da cui vuoi copiare gli oggetti:
   - Per copiare da un bucket generico, scegli la scheda Bucket generici.
   - Per copiare da un bucket di directory, scegli la scheda Directory bucket.
4. Scegliete il bucket generico o il bucket di directory che contiene gli oggetti che desiderate copiare.
5. Scegli la scheda Objects (Oggetti). Nella pagina Oggetti, selezionate la casella di controllo a sinistra dei nomi degli oggetti che desiderate copiare.
6. Nel menu Actions (Operazioni) scegliere Copy (Copia).

   Viene visualizzata la pagina Copia.
7. In Destinazione, scegli Directory bucket per il tipo di destinazione. Per specificare il percorso di destinazione, scegli Browse S3, vai alla destinazione, quindi scegli il pulsante di opzione a sinistra della destinazione. Seleziona Choose destination (Scegli destinazione) nell'angolo in basso a destra.

   In alternativa, immettere il percorso di destinazione.
8. In Checksum, scegli se vuoi copiare gli oggetti con le loro funzioni di checksum esistenti o sostituire le funzioni di checksum esistenti con una nuova. Al momento del caricamento degli oggetti hai la possibilità di specificare l'algoritmo di checksum utilizzato per verificare l'integrità dei dati. Quando effettui la copia dell'oggetto hai la possibilità di scegliere una nuova funzione. Se originariamente non specificate un checksum aggiuntivo, potete utilizzare la sezione Checksum per aggiungerne uno.

> **Note**
>
> Anche se scegli di utilizzare la stessa funzione di checksum, il valore del checksum potrebbe cambiare se l'oggetto ha una dimensione superiore a 16 MB. Il valore del checksum potrebbe cambiare a causa del modo in cui vengono calcolati i checksum per i caricamenti in più parti. Per ulteriori informazioni su come potrebbe cambiare il checksum durante la copia dell'oggetto, consulta Utilizzo di checksum a livello di parte per caricamenti in più parti.

Per modificare la funzione di checksum, scegli Replace with a new checksum function (Sostituisci con una nuova funzione di checksum). Scegliete la nuova funzione di checksum dall'elenco a discesa. Quando l'oggetto viene copiato, il nuovo checksum viene calcolato e memorizzato utilizzando l'algoritmo specificato.

9. Scegli Copy (Copia) nell'angolo in basso a destra. Amazon S3 copia gli oggetti nella destinazione.

Utilizzo della console S3 (copia in un bucket per uso generico)

Per copiare un oggetto da un bucket di directory in un bucket per uso generico

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Scegli la scheda Directory buckets.
4. Scegli il bucket di directory che contiene gli oggetti che desideri copiare.
5. Scegli la scheda Objects (Oggetti). Nella pagina Oggetti, selezionate la casella di controllo a sinistra dei nomi degli oggetti che desiderate copiare.
6. Nel menu Actions (Operazioni) scegliere Copy (Copia).
7. In Destinazione, scegli Secchiello per uso generico per il tipo di destinazione. Per specificare il percorso di destinazione, scegli Browse S3, vai alla destinazione e scegli il pulsante di opzione a sinistra della destinazione. Seleziona Choose destination (Scegli destinazione) nell'angolo in basso a destra.

In alternativa, immettere il percorso di destinazione.

8. In Checksum, scegli se vuoi copiare gli oggetti con le loro funzioni di checksum esistenti o sostituire le funzioni di checksum esistenti con una nuova. Al momento del caricamento degli oggetti hai la possibilità di specificare l'algoritmo di checksum utilizzato per verificare l'integrità dei dati. Quando effettui la copia dell'oggetto hai la possibilità di scegliere una nuova funzione. Se originariamente non avete specificato un checksum aggiuntivo, potete utilizzare la sezione Checksums per aggiungerne uno.

   > **Note**
   >
   > Anche se scegli di utilizzare la stessa funzione di checksum, il valore del checksum potrebbe cambiare se l'oggetto ha una dimensione superiore a 16 MB. Il valore del checksum potrebbe cambiare a causa del modo in cui vengono calcolati i checksum per i caricamenti in più parti. Per ulteriori informazioni su come potrebbe cambiare il checksum durante la copia dell'oggetto, consulta Utilizzo di checksum a livello di parte per caricamenti in più parti.

   Per modificare la funzione di checksum, scegli Replace with a new checksum function (Sostituisci con una nuova funzione di checksum). Scegliete la nuova funzione di checksum dall'elenco a discesa. Quando l'oggetto viene copiato, il nuovo checksum viene calcolato e memorizzato utilizzando l'algoritmo specificato.

9. Scegli Copy (Copia) nell'angolo in basso a destra. Amazon S3 copia gli oggetti nella destinazione.

Utilizzo degli SDK 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);
        }
}
```

Usando il AWS CLI

Il comando di `copy-object` esempio seguente mostra come è possibile utilizzare AWS CLI per copiare un oggetto da un bucket a un altro bucket. È possibile copiare oggetti tra tipi di bucket. Per eseguire questo comando, sostituisci i segnaposto di input dell'utente con le tue informazioni.

```
aws s3api copy-object --copy-source bucket SOURCE_BUCKET/SOURCE_KEY_NAME --
key TARGET_KEY_NAME --bucket TARGET_BUCKET_NAME
```

Per ulteriori informazioni, consulta la sezione copy-object nella Documentazione di riferimento della AWS CLI .

Eliminazione di un oggetto in un bucket di directory

Puoi eliminare oggetti da un bucket di directory Amazon S3 utilizzando la console Amazon S3, AWS Command Line Interface ()AWS CLI o gli SDK. AWS Per ulteriori informazioni, consulta Bucks di directory e Che cos'è S3 Express One Zone?.

> **Warning**
>
> - L'eliminazione di un oggetto non può essere annullata.

- Questa azione elimina tutti gli oggetti specificati. Quando si eliminano le cartelle, attendere che l'azione di eliminazione finisca prima di aggiungere nuovi oggetti alla cartella. In caso contrario, potrebbero essere eliminati anche nuovi oggetti.

> **Note**
>
> Quando eliminate a livello di programmazione più oggetti da un bucket di directory, tenete presente quanto segue:
>
> - Le chiavi degli oggetti nelle richieste `DeleteObjects` devono contenere almeno un carattere diverso dallo spazio. Le stringhe con tutti i caratteri di spazio bianco non sono supportate.
> - Le chiavi oggetto nelle `DeleteObjects` richieste non possono contenere caratteri di controllo Unicode, ad eccezione di newline (`\n`), tab (`\t`) e carriage return (). `\r`

Utilizzo della console S3

Per eliminare oggetti

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Scegli la scheda Bucket di directory.
4. Scegli il bucket di directory che contiene gli oggetti che desideri eliminare.
5. Scegli la scheda Objects (Oggetti). Nell'elenco Oggetti, selezionate la casella di controllo a sinistra dell'oggetto o degli oggetti che desiderate eliminare.
6. Scegli Elimina.
7. Nella pagina Elimina oggetti, immettete **`permanently delete`** nella casella di testo.
8. Scegliere Delete objects (Elimina oggetti).

Utilizzo degli AWS SDK

SDK for Java 2.x

Example

L'esempio seguente elimina gli oggetti in un bucket di directory utilizzando. AWS SDK for Java 2.x

```
static void deleteObject(S3Client s3Client, String bucketName, String objectKey) {



        try {

            DeleteObjectRequest del = DeleteObjectRequest.builder()
                    .bucket(bucketName)
                    .key(objectKey)
                    .build();

            s3Client.deleteObject(del);

            System.out.println("Object " + objectKey + " has been deleted");


        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }

    }
```

SDK for Python

Example

L'esempio seguente elimina gli oggetti in un bucket di directory utilizzando. 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}')
```

Usando il AWS CLI

Il comando di `delete-object` esempio seguente mostra come è possibile utilizzare AWS CLI per eliminare un oggetto da un bucket di directory. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
aws s3api delete-object --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
```

Per ulteriori informazioni, consulta la sezione delete-object nella Documentazione di riferimento della AWS CLI .

Scaricamento di un oggetto in un bucket di directory

I seguenti esempi di codice mostrano come leggere i dati da (scaricare) un oggetto in un bucket di directory Amazon S3 utilizzando l'`GetObject`operazione API.

Utilizzo degli SDK AWS

SDK for Java 2.x

Example

Il seguente esempio di codice mostra come leggere i dati da un oggetto in un bucket di directory utilizzando. 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

Il seguente esempio di codice mostra come leggere i dati da un oggetto in un bucket di directory utilizzando. 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()
```

Usando il AWS CLI

L'esempio `get-object` seguente mostra come utilizzare la AWS CLI per scaricare un oggetto da Amazon S3. Questo comando ottiene l'oggetto *KEY_NAME* dal bucket *bucket-base-name*`--`*azid*`--x-s3` di directory. L'oggetto verrà scaricato in un file denominato *LOCAL_FILE_NAME*. Per eseguire questo comando, sostituire *user input placeholders* con le proprie informazioni.

```
aws s3api get-object --bucket bucket-base-name--azid--x-s3 --
key KEY_NAME LOCAL_FILE_NAME
```

Per ulteriori informazioni, consulta la sezione get-object nella Documentazione di riferimento della AWS CLI .

Utilizzo **`HeadObject`** con i bucket di directory

I seguenti esempi di AWS SDK e AWS CLI mostrano come utilizzare l'operazione API per recuperare `HeadObject` i metadati da un oggetto in un bucket di directory Amazon S3 senza restituire l'oggetto stesso.

Utilizzo degli AWS SDK

SDK for Java 2.x

Example

```
public static void headObject(S3Client s3Client, String bucketName, String
 objectKey) {
    try {
        HeadObjectRequest headObjectRequest = HeadObjectRequest
                .builder()
                .bucket(bucketName)
                .key(objectKey)
                .build();
        HeadObjectResponse response = s3Client.headObject(headObjectRequest);
        System.out.format("Amazon S3 object: \"%s\" found in bucket: \"%s\" with
 ETag: \"%s\"", objectKey, bucketName, response.eTag());
    }
    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
```

Usando il AWS CLI

Il comando di `head-object` esempio seguente mostra come è possibile utilizzare il AWS CLI per recuperare i metadati da un oggetto. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
aws s3api head-object --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
```

Per ulteriori informazioni, consulta la sezione head-object nella Documentazione di riferimento della AWS CLI .

Sicurezza per S3 Express One Zone

Per AWS, la sicurezza del cloud ha la massima priorità. In quanto cliente AWS, puoi trarre vantaggio da un'architettura di data center e di rete progettata per soddisfare i requisiti delle aziende più esigenti a livello di sicurezza. La sicurezza è una responsabilità condivisa tra AWSe l'utente. Il modello di responsabilità condivisa descrive questo come sicurezza del cloud e sicurezza nel cloud:

- La sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che esegue Servizi AWS nel Cloud AWS. AWS fornisce, inoltre, servizi utilizzabili in modo sicuro. I revisori di terze parti testano e verificano periodicamente l'efficacia della sicurezza come parte del AWS Compliance Programs.

  Per ulteriori informazioni sui programmi per la conformità che si applicano ad Amazon S3 Express One Zone, consulta Servizi AWS in Scope by Compliance Program.

- Sicurezza nel cloud: la tua responsabilità è determinata dal Servizio AWS che viene utilizzato. L'utente è anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda e le leggi e le normative applicabili.

Questa documentazione consente di comprendere come applicare il modello di responsabilità condivisa quando si utilizza S3 Express One Zone. I seguenti argomenti illustrano come configurare S3 Express One Zone per soddisfare gli obiettivi di sicurezza e conformità. Inoltre, scoprirai come utilizzare altri Servizi AWS che semplificano il monitoraggio e la protezione delle risorse durante l'utilizzo di S3 Express One Zone.

Argomenti

Protezione e crittografia dei dati

Per ulteriori informazioni su come S3 Express One Zone effettua la crittografia e la protezione dei dati, consulta i seguenti argomenti.

Argomenti

Crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3)

Per impostazione predefinita, tutti gli oggetti archiviati in bucket di directory vengono crittografati automaticamente utilizzando la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3). I caricamenti non crittografati nei bucket di directory non sono consentiti. Per ulteriori informazioni,

consultare Uso della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) e Protezione dei dati con la crittografia.

I bucket di directory non supportano la crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS), la crittografia lato server a doppio livello con chiavi AWS Key Management Service (AWS KMS) (DSSE-KMS) o la crittografia lato server con chiavi di crittografia fornite dal cliente (SSE-C).

Crittografia in transito

L'accesso a S3 Express One Zone è consentito solo tramite HTTPS (TLS).

S3 Express One Zone utilizza endpoint API regionali e zonali. A seconda dell'operazione API Amazon S3 utilizzata, è necessario un endpoint regionale o zonale. È possibile accedere agli endpoint zonali e regionali tramite un endpoint del cloud privato virtuale (VPC) del gateway. L'utilizzo di endpoint gateway non comporta costi supplementari. Per ulteriori informazioni sugli endpoint API regionali e zonali, consulta Servizi di rete per S3 Express One Zone.

Checksum aggiuntivi

S3 Express One Zone offre la possibilità di scegliere l'algoritmo di checksum utilizzato per convalidare i dati durante il caricamento o il download. Puoi selezionare uno dei seguenti algoritmi di controllo dell'integrità dei dati Secure Hash Algorithms (SHA) o Cyclic Redundancy Check (CRC): CRC32, CRC32C, SHA-1 e SHA-256. I checksum basati su MD5 non sono supportati con la classe di storage S3 Express One Zone.

Per ulteriori informazioni, consulta Best practice per il checksum S3 aggiuntivo.

Eliminazione dei dati

Puoi eliminare uno o più oggetti direttamente da S3 Express One Zone utilizzando la console Amazon S3, gli SDK AWS, AWS Command Line Interface (AWS CLI) o la REST API Amazon S3. Tutti gli oggetti nel bucket di directory sono soggetti a costi di archiviazione, pertanto è necessario eliminare gli oggetti non più necessari.

L'eliminazione di un oggetto archiviato in un bucket di directory elimina in modo ricorsivo anche tutte le directory padre, se queste non contengono oggetti diversi dall'oggetto che viene eliminato.

> **Note**
>
> L'eliminazione dell'autenticazione a più fattori (MFA) e la funzione Controllo delle versioni S3 non sono supportati per S3 Express One Zone.

AWS Identity and Access Management (IAM) per S3 Express One Zone

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta gli amministratori a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM controllano chi può essere autenticato (accesso effettuato) e autorizzato (dotato di autorizzazioni) a utilizzare le risorse Amazon S3 in S3 Express One Zone. Puoi utilizzare IAM senza alcun costo aggiuntivo.

Per impostazione predefinita, gli utenti non dispongono delle autorizzazioni per i bucket di directory e le operazioni S3 Express One Zone. Per concedere le autorizzazioni di accesso per i bucket di directory, puoi utilizzare IAM per creare utenti, gruppi o ruoli e collegare le autorizzazioni a tali identità. Per ulteriori informazioni su IAM, consulta Best Practice per la sicurezza in IAM nella Guida per l'utente di IAM.

Per fornire l'accesso, puoi aggiungere autorizzazioni a utenti, gruppi o ruoli tramite i mezzi seguenti:

- Utenti e gruppi in AWS IAM Identity Center: crea un set di autorizzazioni. Segui le istruzioni riportate nella pagina Create a permission set (Creazione di un set di autorizzazioni) nella Guida per l'utente di AWS IAM Identity Center .
- Utenti gestiti in IAM tramite un provider di identità: crea un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina Creating a role for a third-party identity provider (federation) (Creazione di un ruolo per un provider di identità di terze parti [federazione]) nella Guida per l'utente di IAM.
- Ruoli e utenti IAM: crea un ruolo che l'utente è in grado di assumere. Segui le istruzioni in Creazione di un ruolo per delegare le autorizzazioni a un utente IAM nella Guida per l'utente di IAM.

Per impostazione predefinita, i bucket di directory sono privati e l'accesso è possibile solo dagli utenti a cui è concesso esplicitamente l'accesso. Il limite di controllo degli accessi per i bucket di directory è impostato solo a livello di bucket. Al contrario, il limite di controllo degli accessi per i bucket per uso generico può essere impostato a livello di bucket, prefisso o tag dell'oggetto. Questa differenza

significa che i bucket di directory sono l'unica risorsa che puoi includere nelle policy dei bucket o nelle policy di identità IAM per l'accesso a S3 Express One Zone.

Con S3 Express One Zone, oltre all'autorizzazione IAM, vengono autenticate e autorizzate le richieste tramite un nuovo meccanismo basato sulla sessione gestito dall'operazione API `CreateSession`. Puoi utilizzare `CreateSession` per richiedere credenziali temporanee che forniscono un accesso a bassa latenza al bucket. Queste credenziali temporanee sono definite per un bucket di directory specifico.

Per utilizzarlo`CreateSession`, ti consigliamo di utilizzare la versione più recente degli AWS SDK o di utilizzare AWS Command Line Interface (AWS CLI). Gli AWS SDK supportati e gli SDK AWS CLI gestiscono la creazione, l'aggiornamento e la chiusura della sessione per tuo conto.

Utilizza i token di sessione solo con operazioni (a livello di oggetto) zonali (fatta eccezione per `CopyObject` e `HeadBucket`) per distribuire la latenza associata all'autorizzazione su un determinato numero di richieste in una sessione. Per operazioni API degli endpoint regionali (operazioni a livello di bucket), viene utilizzata l'autorizzazione IAM, che non prevede la gestione di una sessione. Per ulteriori informazioni, consultare AWS Identity and Access Management (IAM) per S3 Express One Zone e Autorizzazione `CreateSession`.

Per ulteriori informazioni su IAM per S3 Express One Zone, consulta i seguenti argomenti.

Argomenti

- Principali
- Risorse
- Azioni per S3 Express One Zone
- Chiavi di condizione per S3 Express One Zone
- Come vengono autorizzate e autenticate le operazioni API

Principali

Quando si crea una policy basata sulle risorse per concedere l'accesso ai bucket, è necessario utilizzare l'elemento `Principal` per specificare la persona o l'applicazione che può effettuare una richiesta per un'azione o un'operazione su tale risorsa. Per le policy dei bucket di directory, puoi utilizzare i seguenti principali:

- Un account AWS
- Un utente IAM

- Un ruolo IAM:
- Un utente federato

Per ulteriori informazioni, consulta la sezione Principal nella Guida per l'utente di IAM.

Risorse

Gli Amazon Resource Names (ARN) per i bucket di directory contengono lo spazio dei `s3express` nomi Regione AWS, l'ID dell' AWS account e il nome del bucket di directory, che include l'ID della zona di disponibilità. Per accedere ed eseguire azioni sul bucket di directory, è necessario utilizzare il seguente formato ARN:

```
arn:aws:s3express:region:account-id:bucket/base-bucket-name--azid--x-s3
```

Per ulteriori informazioni sugli ARN, consulta Amazon Resource Names (ARNs) nella Guida per l'utente di IAM. Per ulteriori informazioni sulle risorse, consulta Elementi delle policy JSON IAM: Resource nella Guida per l'utente di IAM.

Azioni per S3 Express One Zone

In una policy IAM basata sull'identità o una policy basata sulle risorse, vengono definite quali azioni S3 sono consentite o negate. Le azioni S3 Express One Zone corrispondono a operazioni API specifiche. S3 Express One Zone dispone di uno spazio dei nomi IAM univoco, distinto dallo spazio dei nomi standard di Amazon S3. Questo spazio dei nomi è `s3express`.

Quando si concede l'autorizzazione `s3express:CreateSession`, l'operazione API `CreateSession` è in grado di recuperare i token di sessione durante l'accesso alle operazioni API (o a livello di oggetto) degli endpoint zonali. Questi token di sessione restituiscono le credenziali utilizzate per concedere l'accesso a tutte le altre operazioni API degli endpoint zonali. Di conseguenza, non è necessario concedere le autorizzazioni di accesso alle operazioni API zonali utilizzando le policy IAM. Invece, il token di sessione consente l'accesso.

Per ulteriori informazioni sulle operazioni API degli endpoint zonali e regionali, consulta Servizi di rete per S3 Express One Zone. Per ulteriori informazioni sulle operazioni API `CreateSession`, consulta CreateSession nella Documentazione di riferimento delle API Amazon Simple Storage Service.

Puoi specificare le seguenti operazioni nell'elemento `Action` di un'istruzione di policy IAM. Utilizza le policy per concedere le autorizzazioni per eseguire un'operazione in AWS. Quando si utilizza un'azione in una policy, in genere si consente o si nega l'accesso all'operazione API con lo stesso

nome. Tuttavia, in alcuni casi, una singola azione controlla l'accesso a più operazioni API. L'accesso alle azioni a livello di bucket può essere concesso solo nelle policy basate sulle identità IAM (utente o ruolo) e non nelle policy dei bucket.

Azioni e chiavi di condizione per S3 Express One Zone

Azione	API	Descrizione	Livello di accesso	Chiavi di condizione
`s3express:CreateBucket`	`CreateBucket`	Concede l'autorizzazione per creare un nuovo bucket.	Scrittura	`s3express:authType` `s3express:LocationName` `s3express:ResourceAccount` `s3express:signatureversion` `s3express:TlsVersion` `s3express:x-amz-content-sha256`
`s3express:CreateSession`	`CreateSession`	Concede l'autorizzazione per creare un token di sessione, che è utilizzato per concedere l'accesso a tutte le operazioni API (a livello di oggetto) zonali, come	Scrittura	`s3express:authType` `s3express:SessionMode`

Azione	API	Descrizione	Livello di accesso	Chiavi di condizione
		`PutObject` , `GetObject` e così via.		`s3express:ResourceAccount` `s3express:signatureversion` `s3express:signatureAge` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Azione	API	Descrizione	Livello di accesso	Chiavi di condizione
`s3express:DeleteBucket`	`DeleteBucket`	Concede l'autorizzazione per eliminare il bucket denominato nell'URI.	Scrittura	`s3express:authType` `s3express:ResourceAccount` `s3express:signatureversion` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Azione	API	Descrizione	Livello di accesso	Chiavi di condizione
s3express:DeleteBucketPolicy	DeleteBucketPolicy	Concede l'autorizzazione per eliminare la policy su un bucket specificato.	Gestione delle autorizzazioni	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Azione	API	Descrizione	Livello di accesso	Chiavi di condizione
s3express:GetBucketPolicy	GetBucketPolicy	Concede l'autorizzazione per restituire la policy del bucket specificato.	Lettura	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Azione	API	Descrizione	Livello di accesso	Chiavi di condizione
`s3express:ListAllMyDirectoryBuckets`	`ListDirectoryBuckets`	Concede l'autorizzazione per elencare tutti i bucket di directory di proprietà del mittente autenticato della richiesta.	Elenco	`s3express:authType` `s3express:ResourceAccount` `s3express:signatureversion` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Azione	API	Descrizione	Livello di accesso	Chiavi di condizione
`s3express:PutBucketPolicy`	`PutBucketPolicy`	Concede l'autorizzazione per aggiungere o sostituire una policy del bucket in un bucket.	Gestione delle autorizzazioni	`s3express:authType` `s3express:ResourceAccount` `s3express:signatureversion` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Chiavi di condizione per S3 Express One Zone

S3 Express One Zone definisce le seguenti chiavi di condizione che possono essere utilizzate nell'elemento `Condition` di una policy IAM. Puoi utilizzare queste chiavi per perfezionare ulteriormente le condizioni in base alle quali si applica l'istruzione di policy.

Chiave di condizione	Descrizione	Type
`s3express:authType`	Filtra l'accesso in base al metodo di autenticazione. Per limitare le richieste in arrivo all'utilizzo di un metodo di autenticazione specifico, puoi utilizzare questa chiave di condizione opzionale. Ad esempio, puoi utilizzare questa chiave di	Stringa

Chiave di condizione	Descrizione	Type
	condizione per consentire solo l'intestazione `Authorization` HTTP da utilizzare nell'autenticazione della richiesta. Valori validi: `REST-HEADER` , `REST-QUERY-STRING`	
`s3express:LocationName`	Filtra l'accesso all'operazione API `CreateBucket` in base a un ID zona di disponibilità specifico, ad esempio, `usw2-az1`. Valore di esempio: `usw2-az1`	Stringa
`s3express:ResourceAccount`	Filtra l'accesso in base all'ID del proprietario della risorsa. Account AWS Per limitare l'accesso di utenti, ruoli o applicazioni ai bucket di directory di proprietà di un Account AWS ID specifico, puoi utilizzare la chiave di `s3express:ResourceAccount` condizione `aws:ResourceAccount` o. Puoi utilizzare questa chiave di condizione nelle policy di identità AWS Identity and Access Management (IAM) o nelle policy degli endpoint del cloud privato virtuale (VPC). Ad esempio, puoi utilizzare questa chiave di condizione per impedire ai client all'interno del tuo VPC di accedere a bucket che non possiedi. Valore di esempio: `111122223333`	Stringa

Chiave di condizione	Descrizione	Type
`s3express:SessionMode`	Filtra l'accesso in base all'autorizzazione richiesta dall'operazione API `CreateSession` . Per impostazione predefinita, la sessione è `ReadWrite` . Puoi utilizzare questa chiave di condizione per limitare l'accesso a `ReadOnly` o per rifiutare esplicitamente l'accesso `ReadWrite` . Per ulteriori informazioni, consulta Esempi di policy dei bucket di directory per S3 Express One Zone e CreateSession nella Documentazione di riferimento delle API Amazon Simple Storage Service. Valori validi: `ReadWrite` , `ReadOnly`	Stringa
`s3express:signatureAge`	Filtra l'accesso in base all'età in millisecondi della firma della richiesta. Questa condizione è valida solo per gli URL prefirmati. Nella versione 4 di AWS Signature, la chiave di firma è valida per un massimo di sette giorni. Pertanto, anche le firme sono valide per un massimo di sette giorni. Per ulteriori informazioni, consulta Introduzione alla firma delle richieste nella Documentazione di riferimento delle API di Amazon Simple Storage Service. Puoi utilizzare questa condizione per limitare ulteriormente la durata della firma. Valore di esempio: `600000`	Numerico

Chiave di condizione	Descrizione	Type
`s3express:signaturversion`	Identifica la versione di AWS Signature che desideri supportare per le richieste autentica te. Per le richieste autenticate, S3 Express One Zone supporta Signature Version 4. Valore valido: `"AWS4-HMAC-SHA256"` (identifica la versione 4 di Signature)	Stringa
`s3express:TlsVersion`	Filtra l'accesso in base alla versione TLS utilizzata dal client. Puoi utilizzare la chiave di `s3:TlsVersion` condizione per scrivere policy IAM, Virtual Private Cloud Endpoint (VPCE) o bucket che limitano l'accesso di utenti o applicazioni ai bucket di directory in base alla versione TLS utilizzata dal client. Puoi anche utilizzare questa chiave di condizione per scrivere policy che richiedono una versione TLS minima. Valore di esempio: `1.3`	Numerico

Chiave di condizione	Descrizione	Type
`s3express:x-amz-content-sha256`	Filtra l'accesso in base ai contenuti non firmati nel bucket. Questa chiave di condizione può essere utilizzata per non consentire contenuti non firmati nel bucket. Quando si utilizza Signature Version 4, per le richieste che utilizzano l'intestazione `Authorization` , viene aggiunta l'intestazione `x-amz-content-sha256` nel calcolo della firma e quindi impostato il relativo valore sul payload hash. Puoi utilizzare questa chiave di condizione nella policy del bucket per rifiutare qualsiasi caricamento in cui i payload non sono firmati. Per esempio: • Nega i caricamenti che utilizzano l'intestazione `Authorization` per autenticare le richieste ma non firmare il payload. Per ulteriori informazioni, consulta Trasferimento del carico utile in un unico blocco nella Documentazione di riferimento delle API di Amazon Simple Storage Service. • Nega i caricamenti che utilizzano URL predefiniti. Gli URL prefirmati hanno sempre un `UNSIGNED_PAYLOAD` . Per ulteriori informazioni, consulta la sezione Autenticazione delle richieste e Metodi di autenticazione nella Documentazione di riferimento delle API di Amazon Simple Storage Service. Valore valido: `UNSIGNED-PAYLOAD`	Stringa

Come vengono autorizzate e autenticate le operazioni API

Nella tabella seguente vengono elencate le informazioni di autorizzazione e autenticazione per le operazioni API S3 Express One Zone. Per ogni operazione API, la tabella mostra il nome dell'operazione API, l'azione IAM, il tipo di endpoint (regionale o zonale) e il meccanismo di autorizzazione (IAM o basato sulla sessione). In questa tabella viene indicato anche dove è supportato l'accesso multi-account. L'accesso alle azioni a livello di bucket può essere concesso solo nelle policy basate sull'identità IAM (utente o ruolo) e non nelle policy dei bucket.

API	Tipo di endpoint	Operazione IAM	Accesso multi-account
`CreateBucket`	Regionale	`s3express:CreateBucket`	No
`DeleteBucket`	Regionale	`s3express:DeleteBucket`	No
`ListDirectoryBuckets`	Regionale	`s3express:ListAllMyDirectoryBuckets`	No
`PutBucketPolicy`	Regionale	`s3express:PutBucketPolicy`	No
`GetBucketPolicy`	Regionale	`s3express:GetBucketPolicy`	No
`DeleteBucketPolicy`	Regionale	`s3express:DeleteBucketPolicy`	No
`CreateSession`	Zonale	`s3express:CreateSession`	Sì
`CopyObject`	Zonale	`s3express:CreateSession`	Sì
`DeleteObject`	Zonale	`s3express:CreateSession`	Sì
`DeleteObjects`	Zonale	`s3express:CreateSession`	Sì
`HeadObject`	Zonale	`s3express:CreateSession`	Sì
`PutObject`	Zonale	`s3express:CreateSession`	Sì

API	Tipo di endpoint	Operazione IAM	Accesso multi-account
`GetObjectAttributes`	Zonale	`s3express:CreateSession`	Sì
`ListObjectsV2`	Zonale	`s3express:CreateSession`	Sì
`HeadBucket`	Zonale	`s3express:CreateSession`	Sì
`CreateMultipartUpload`	Zonale	`s3express:CreateSession`	Sì
`UploadPart`	Zonale	`s3express:CreateSession`	Sì
`UploadPartCopy`	Zonale	`s3express:CreateSession`	Sì
`CompleteMultipartUpload`	Zonale	`s3express:CreateSession`	Sì
`AbortMultipartUpload`	Zonale	`s3express:CreateSession`	Sì
`ListParts`	Zonale	`s3express:CreateSession`	Sì
`ListMultipartUploads`	Zonale	`s3express:CreateSession`	Sì

Policy basate sulle identità IAM per S3 Express One Zone

Prima di poter creare bucket di directory o utilizzare la classe di archiviazione Amazon S3 Express One Zone, devi concedere le autorizzazioni necessarie al ruolo o agli utenti AWS Identity and Access Management (IAM). Questa policy di esempio consente l'accesso all'operazione API `CreateSession` (per l'utilizzo con le operazioni API [a livello di oggetto] degli endpoint zonali) e a tutte le operazioni API (a livello di bucket) degli endpoint regionali. Questa policy consente l'operazione API `CreateSession` per l'utilizzo con tutti i bucket di directory, ma le operazioni API

degli endpoint regionali sono consentite solo per l'utilizzo con il bucket di directory specificato. Per utilizzare questa policy di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
{
     "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": "*"
         }
     ]
 }
```

Esempi di policy dei bucket di directory per S3 Express One Zone

In questa sezione vengono fornite policy dei bucket di directory di esempio per l'utilizzo con la classe di archiviazione Amazon S3 Express One Zone. Per usare queste policy, sostituisci *user input placeholders* con le tue informazioni.

La seguente policy del bucket di esempio consente all'ID dell'Account AWS *111122223333* di utilizzare l'operazione API `CreateSession` con la sessione `ReadWrite` predefinita per il bucket di directory specificato. Questa policy concede l'accesso alle operazioni API (a livello di oggetto) degli endpoint zonali.

Example – Policy del bucket per consentire chiamate **`CreateSession`** con la sessione **`ReadWrite`** predefinita

```
{
    "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 – Policy del bucket per consentire chiamate **`CreateSession`** con una sessione **`ReadOnly`**

La seguente policy del bucket di esempio consente all'ID Account AWS dell'*111122223333* di utilizzare l'operazione API `CreateSession`. Questa policy utilizza la chiave di condizione `s3express:SessionMode` con il valore `ReadOnly` per impostare una sessione di sola lettura.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadOnlyAccess",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
            "Action": "s3express:CreateSession",
            "Resource": "*",
```

```
            "Condition": {
                "StringEquals": {
                    "s3express:SessionMode": "ReadOnly"
                }
            }
        }
    ]
}
```

Example – Policy del bucket per consentire accesso multi-account per chiamate **`CreateSession`**

La seguente policy del bucket di esempio consente all'Account AWSID dell'*111122223333* di utilizzare l'operazione API `CreateSession` per il bucket di directory specificato di proprietà dell'ID dell'Account 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"

        }
    ]
}
```

Autorizzazione **`CreateSession`**

Amazon S3 Express One Zone supporta l'autorizzazione AWS Identity and Access Management (AWS IAM) e l'autorizzazione basata sulla sessione:

- Per utilizzare le operazioni dell'API degli endpoint regionali (operazioni a livello di bucket o piano di controllo) con S3 Express One Zone, utilizzi il modello di autorizzazione IAM, che non prevede la gestione delle sessioni. Le autorizzazioni sono concesse per le singole azioni. Per ulteriori informazioni, consulta AWS Identity and Access Management (IAM) per S3 Express One Zone.
- Per utilizzare le operazioni API degli endpoint zonali (operazioni a livello di oggetto o piano dati), si utilizza l'operazione API `CreateSession` per creare e gestire sessioni ottimizzate per l'autorizzazione a bassa latenza delle richieste di dati. Per recuperare e utilizzare un token di sessione, occorre consentire l'azione `s3express:CreateSession` per il bucket di directory in una policy basata sull'identità o in una policy di bucket. Per ulteriori informazioni, consulta AWS Identity and Access Management (IAM) per S3 Express One Zone. Se si accede a S3 Express One Zone nella console Amazon S3, tramite AWS Command Line Interface (AWS CLI) o mediante gli SDK AWS, S3 Express One Zone crea una sessione per conto dell'utente.

Se si utilizza la REST API Amazon S3, è possibile quindi utilizzare l'operazione API `CreateSession` per ottenere credenziali di sicurezza temporanee che includono un ID della chiave di accesso, una chiave di accesso segreta, un token di sessione e una data di scadenza. Le credenziali temporanee forniscono le stesse autorizzazioni delle credenziali di sicurezza a lungo termine, come le credenziali degli utenti IAM, ma le credenziali di sicurezza temporanee devono includere un token di sessione.

Modalità sessione

Modalità sessione definisce l'ambito della sessione. Nella policy di bucket, puoi specificare la chiave di condizione `s3express:SessionMode` per controllare chi può creare una sessione `ReadWrite` o `ReadOnly`. Per ulteriori informazioni sulle sessioni `ReadWrite` o `ReadOnly`, consulta il parametro `x-amz-create-session-mode` per CreateSession nella Documentazione di riferimento delle API Amazon S3. Per ulteriori informazioni sulla policy di bucket da creare, consulta Esempi di policy dei bucket di directory per S3 Express One Zone.

Token di sessione

Quando effettui una chiamata utilizzando le credenziali di sicurezza temporanee, la chiamata deve includere un token di sessione. Il token di sessione viene restituito insieme alle credenziali temporanee. L'ambito di un token di sessione viene definito dal bucket di directory e il token di sessione viene utilizzato per verificare che le credenziali di sicurezza siano valide e non siano scadute. Per proteggere le sessioni, le credenziali di sicurezza temporanee scadono dopo 5 minuti.

`CopyObject` e `HeadBucket`

L'ambito delle credenziali di sicurezza temporanee viene definito da un bucket di directory specifico e le credenziali vengono abilitate automaticamente per tutte le chiamate API operative zonali (a livello di oggetto) verso un bucket di directory specifico. A differenza di altre operazioni API degli endpoint zonali, `CopyObject` e `HeadBucket` non utilizzano l'autenticazione `CreateSession`. Tutte le richieste `CopyObject` e `HeadBucket` devono essere autenticate e firmate utilizzando credenziali IAM. Tuttavia, `CopyObject` e `HeadBucket` sono ancora autorizzate da `s3express:CreateSession`, come altre operazioni API degli endpoint zonali.

Per ulteriori informazioni, consulta CreateSession in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Best practice per la sicurezza di S3 Express One Zone

Amazon S3 Express One Zone fornisce una serie di funzionalità di sicurezza che occorre valutare durante lo sviluppo e l'implementazione delle proprie policy di sicurezza. Le seguenti best practice sono linee guida generali e non rappresentano una soluzione di sicurezza completa. Poiché queste best practice potrebbero non essere appropriate o sufficienti per il tuo ambiente, considerale come consigli utili più che prescrizioni.

Impostazioni predefinite di Blocco dell'accesso pubblico e Proprietà dell'oggetto

Per utilizzare la classe di archiviazione S3 Express One Zone, devi utilizzare un bucket di directory S3. I bucket di directory supportano Blocco dell'accesso pubblico S3 e Proprietà dell'oggetto S3. Queste funzionalità S3 vengono utilizzate per la verifica e la gestione dell'accesso ai bucket e agli oggetti.

Per impostazione predefinita, tutte impostazioni Blocco dell'accesso pubblico per i nuovi bucket sono abilitate. Inoltre, Proprietà dell'oggetto è impostata su Proprietario del bucket applicato, il che significa che le liste di controllo degli accessi (ACL) sono disabilitate. Queste impostazioni non possono essere modificate. Per ulteriori informazioni su queste funzionalità, consulta the section called “Blocco dell'accesso pubblico” e the section called “Controllo della proprietà degli oggetti”.

> **Note**
>
> Non è possibile concedere l'accesso agli oggetti archiviati nei bucket di directory, ma solo ai bucket di directory. Il modello di autorizzazione per S3 Express One Zone è diverso dal modello di autorizzazione per Amazon S3. Per ulteriori informazioni, consulta Autorizzazione `CreateSession`.

Autenticazione e autorizzazione

I meccanismi di autenticazione e autorizzazione per S3 Express One Zone differiscono a seconda che si stia effettuando richieste alle operazioni API degli endpoint zonali o alle operazioni API degli endpoint regionali. Le operazioni API zonali sono operazioni a livello di oggetto (piano dati). Le operazioni API regionali sono operazioni a livello di bucket (piano di controllo (control-plane)).

Con S3 Express One Zone, puoi autenticare e autorizzare le richieste alle operazioni API degli endpoint zonali tramite un nuovo meccanismo basato sulla sessione ottimizzato per fornire la latenza minima. Con l'autenticazione basata sulla sessione, gli SDK AWS utilizzano l'operazione API `CreateSession` per richiedere credenziali temporanee che forniscono un accesso a bassa latenza al bucket di directory. Queste credenziali temporanee sono definite per un bucket di directory specifico e scadono dopo 5 minuti. È possibile utilizzare queste credenziali temporanee per firmare chiamate API (a livello di oggetto) zonali. Per ulteriori informazioni, consulta Autorizzazione `CreateSession`.

Richieste di firma con credenziali S3 Express One Zone

Le credenziali S3 Express One Zone vengono utilizzate per firmare le richieste API (a livello di oggetto) degli endpoint zonali con AWS Signature Version 4, con `s3express` come nome di servizio. Quando si firmano le richieste, viene utilizzata la chiave segreta restituita da `CreateSession` e viene fornito anche il token di sessione con `x-amzn-s3session-token header`. Per ulteriori informazioni, consulta CreateSession.

Gli SDK AWS supportati per la classe S3 Express One Zone gestiscono le credenziali e la firma per conto dell'utente. È consigliabile utilizzare gli SDK AWS per S3 Express One Zone per aggiornare le credenziali e firmare automaticamente le richieste.

Richieste di firma con credenziali IAM

Tutte le chiamate API (a livello di bucket) regionali devono essere autenticate e firmate da credenziali AWS Identity and Access Management (IAM) anziché da credenziali di sessione temporanee. Le credenziali IAM sono costituite dall'ID chiave di accesso e dalla chiave di accesso segreta per le identità IAM. Tutte le richieste `CopyObject` e `HeadBucket` devono essere autenticate e firmate utilizzando credenziali IAM.

Per ottenere la latenza minima per le chiamate operative (a livello di oggetto) zonali, si consiglia di utilizzare le credenziali S3 Express One Zone ottenute dalla chiamata `CreateSession` per firmare le richieste, fatta eccezione per le richieste a `CopyObject` e `HeadBucket`.

Utilizzare AWS CloudTrail

AWS CloudTrail offre un record delle azioni eseguite da un utente, un ruolo o un Servizio AWS in Amazon S3. È possibile utilizzare le informazioni raccolte da CloudTrail per determinare quanto segue:

- La richiesta effettuata ad Amazon S3
- L'indirizzo IP dal quale è stata effettuata la richiesta
- L'utente che ha effettuato la richiesta
- L'ora in cui è stata effettuata la richiesta
- Dettagli aggiuntivi relativi alla richiesta

Quando si configura ilAccount AWS, CloudTrail è abilitato per impostazione predefinita. Vengono registrate le seguenti operazioni API regionali degli endpoint (operazioni API a livello di bucket o piano di controllo). CloudTrail

- `CreateBucket`
- `DeleteBucket`
- `DeleteBucketPolicy`
- `PutBucketPolicy`
- `GetBucketPolicy`
- `ListDirectoryBuckets`

È possibile visualizzare gli eventi recenti nella console. CloudTrail Per creare un record continuo di attività ed eventi per i tuoi bucket Amazon S3, puoi creare un percorso nella console. CloudTrail Per ulteriori informazioni, consulta Creating a trail nella Guida per l'utente di AWS CloudTrail.

> **Note**
>
> Per S3 Express One Zone, la CloudTrail registrazione delle operazioni API degli endpoint zonali (a livello di oggetto o piano dati) (ad esempio o) non è supportata. `PutObject` `GetObject`

Implementazione del monitoraggio mediante gli strumenti di monitoraggio AWS

Il monitoraggio è importante per mantenere l'affidabilità, la sicurezza, la disponibilità e le prestazioni di Amazon S3 e delle soluzioni AWS. AWS offre vari strumenti e servizi che consentono di monitorare Amazon S3 e gli altri Servizi AWS. Ad esempio, puoi monitorare i CloudWatch parametri di Amazon per Amazon S3, in particolare i parametri `NumberOfObjects` e `BucketSizeBytes` i parametri di storage.

Gli oggetti archiviati nella classe di archiviazione S3 Express One Zone non verranno riflessi nelle metriche di archiviazione `BucketSizeBytes` e `NumberOfObjects` per Amazon S3. Tuttavia, le metriche di archiviazione `BucketSizeBytes` e `NumberOfObjects` sono supportate per S3 Express One Zone. Per visualizzare le metriche preferite, puoi distinguere tra le classi di archiviazione Amazon S3 e la classe di archiviazione S3 Express One Zone specificando una dimensione `StorageType`. Per ulteriori informazioni, consulta Monitoraggio delle metriche con Amazon CloudWatch.

Per ulteriori informazioni, consultare Monitoraggio delle metriche con Amazon CloudWatch e Monitoraggio di Amazon S3.

Ottimizzazione delle prestazioni di Amazon S3 Express One Zone

Amazon S3 Express One Zone è una classe di archiviazione S3 a singola zona di disponibilità (AZ), ad alte prestazioni, creata appositamente per fornire un accesso coerente ai dati di pochi millisecondi per le applicazioni sensibili alla latenza. S3 Express One Zone è la prima classe di archiviazione S3 che offre la possibilità di co-ubicare archiviazione oggetti ad alte prestazioni e risorse di calcolo AWS, come Amazon Elastic Compute Cloud, Amazon Elastic Kubernetes Service e Amazon Elastic Container Service, all'interno di una singola zona di disponibilità. La co-ubicazione delle risorse di archiviazione e calcolo ottimizza le prestazioni di calcolo e i costi e fornisce una maggiore velocità di elaborazione dei dati.

S3 Express One Zone fornisce elasticità delle prestazioni simile a quella delle altre classi di archiviazione S3, ma con latenze coerenti di richiesta di lettura e scrittura per il primo byte di pochi millisecondi, fino a 10 volte più rapidamente di S3 Standard. S3 Express One Zone è progettato da zero per supportare un velocità di trasmissione effettiva ottimale fino a livelli di aggregazione molto elevati. La classe di archiviazione S3 Express One Zone utilizza un'architettura personalizzata per ottimizzare le prestazioni e offrire una latenza di richiesta costantemente bassa archiviando i dati su hardware ad alte prestazioni. Il protocollo a oggetti per S3 Express One Zone è stato migliorato per ottimizzare l'autenticazione e il sovraccarico di metadati.

Per aumentare ulteriormente la velocità di accesso e supportare centinaia di migliaia di richieste al secondo, S3 Express One Zone archivia i dati in un nuovo tipo di bucket: un bucket di directory Amazon S3. Ogni bucket di directory S3 può supportare centinaia di migliaia di transazioni al secondo (TPS).

La combinazione di hardware e software dedicati ad alte prestazioni che forniscono una velocità di accesso ai dati di pochi millisecondi e bucket di directory in grado di dimensionarsi per un numero elevato di transazioni al secondo, rende S3 Express One Zone la classe di archiviazione Amazon S3 migliore per operazioni con un elevato numero di richieste o applicazioni con prestazioni critiche.

Nei seguenti argomenti vengono descritte le linee guida sulle best practice e i modelli di progettazione per l'ottimizzazione delle prestazioni con applicazioni che utilizzano la classe di archiviazione S3 Express One Zone.

Argomenti

Linee guida sulle prestazioni e modelli di progettazione per S3 Express One Zone

Durante la creazione di applicazioni che caricano e recuperano oggetti da Amazon S3 Express One Zone, segui le nostre linee guida sulle best practice per ottimizzare le prestazioni. Per utilizzare la classe di archiviazione S3 Express One Zone, devi creare una directory di bucket S3. La classe di archiviazione S3 Express One Zone non è supportata per l'utilizzo con bucket per uso generico S3.

Per le linee guida sulle prestazioni per tutte le altre classi di archiviazione Amazon S3 e i bucket per uso generico S3, consulta Modelli di progettazione delle best practice: ottimizzazione delle prestazioni di Amazon S3.

Per ottenere prestazioni ottimali per l'applicazione quando si utilizzano la classe di archiviazione S3 Express One Zone e i bucket di directory, è opportuno seguire le linee guida e i modelli di progettazione.

Argomenti

- Utilizzo dell'autenticazione basata sulla sessione
- Best practice per il checksum S3 aggiuntivo
- Utilizzo della versione più recente degli SDK AWS e delle librerie di runtime comuni
- Risoluzione dei problemi relativi alle prestazioni

Co-ubicazione dello spazio di archiviazione S3 Express One Zone con risorse di calcolo AWS

Ogni bucket di directory viene archiviato in una singola zona di disponibilità selezionata al momento della creazione del bucket. Puoi iniziare creando un nuovo bucket di directory in una zona di disponibilità locale nei carichi di lavoro o nelle risorse di calcolo. Quindi, puoi iniziare immediatamente letture e scritture a latenza molto bassa. I bucket di directory sono i primi bucket S3 in cui è possibile scegliere la zona di disponibilità in una Regione AWS per ridurre la latenza tra calcolo e archiviazione.

Se accedi a bucket di directory tra zone di disponibilità, la latenza aumenterà. Per ottimizzare le prestazioni, ti consigliamo di accedere a un bucket di directory dalle istanze di Amazon Elastic Container Service, Amazon Elastic Kubernetes Service e Amazon Elastic Compute Cloud che si trovano nella stessa zona di disponibilità, se possibile.

Bucket di directory

Ogni bucket di directory può supportare centinaia di migliaia di transazioni al secondo (TPS).
A differenza dei bucket per uso generico, i bucket di directory organizzano le chiavi in maniera gerarchica in directory anziché prefissi. Un prefisso è una stringa di caratteri all'inizio del nome della chiave dell'oggetto. Puoi pensare ai prefissi come un modo per organizzare i dati in modo simile alle directory. Tuttavia, i prefissi non sono directory.

I prefissi organizzano i dati in un spazio dei nomi semplice all'interno di bucket per uso generico e non esistono limiti al numero di prefissi all'interno di un bucket per uso generico. Ogni prefisso può raggiungere almeno 3.500/`DELETE`o 5.500 `PUT` `POST`/`GET`richieste al secondo. `HEAD` Puoi anche parallelizzare le richieste su più prefissi per dimensionare le prestazioni. Tuttavia, questo dimensionamento, nel caso di operazioni di lettura e scrittura, avviene gradualmente e non è istantaneo. Sebbene i bucket per uso generico eseguano il dimensionamento alla nuova frequenza di richiesta più elevata, si potrebbero verificare alcuni errori con codice di stato HTTP 503 (Service Unavailable).

Con uno spazio dei nomi gerarchico, il delimitatore nella chiave dell'oggetto è importante. Il solo delimitatore supportato è una barra (`/`). Le directory sono determinate dai limiti dei delimitatori. Ad esempio, la chiave dell'oggetto `dir1/dir2/file1.txt` comporta che le directory `dir1/` e `dir2/` vengano create automaticamente e che l'oggetto `file1.txt` venga aggiunto alla directory `/dir2` nel percorso `dir1/dir2/file1.txt`.

Le directory create quando gli oggetti vengono caricati nei bucket di directory non dispongono di limiti TPS per prefisso e vengono pre-dimensionate automaticamente per ridurre la possibilità di errori HTTP 503 (Service Unavailable). Questo dimensionamento automatico consente alle applicazioni di parallelizzare le richieste di lettura e scrittura all'interno e tra le directory in base alle esigenze.

Parallelizzazione delle richieste di dimensionamento orizzontale dei bucket di directory

Puoi ottenere prestazioni ottimali inviando più richieste simultanee ai bucket di directory per distribuire le richieste su connessioni separate per massimizzare la larghezza di banda accessibile. S3 Express One Zone non impone limiti al numero di connessioni effettuate al bucket di directory. Le singole directory possono dimensionare le prestazioni orizzontalmente e automaticamente quando si verifica un numero elevato di scritture simultanee nella stessa directory.

Quando una chiave dell'oggetto viene inizialmente creata e il relativo nome della chiave include una directory, la directory viene creata automaticamente per l'oggetto. I successivi caricamenti di oggetti nella stessa directory non richiedono la creazione della directory, riducendo pertanto la latenza su caricamenti di oggetti nelle directory esistenti.

Sebbene l'archiviazione di oggetti all'interno di un bucket di directory supporti entrambe le strutture di directory superficiali e profonde, i bucket di directory eseguono automaticamente il dimensionamento orizzontale, con una latenza inferiore sui caricamenti simultanei nella stessa directory o negli elementi di pari livello delle directory parallele.

Utilizzo dell'autenticazione basata sulla sessione

S3 Express One Zone e i bucket di directory supportano un nuovo meccanismo di autorizzazione basato sulla sessione per autenticare e autorizzare le richieste a un bucket di directory. Con l'autenticazione basata sulla sessione, gli SDK AWS utilizzano automaticamente l'operazione API `CreateSession` per creare un token di sessione temporaneo che può essere utilizzato per l'autorizzazione a bassa latenza delle richieste di dati in un bucket di directory.

Gli SDK AWS utilizzano l'operazione API `CreateSession` per richiedere credenziali temporanee, quindi creano e aggiornano automaticamente i token per conto dell'utente ogni 5 minuti. Per sfruttare i vantaggi in termini di prestazioni della classe di archiviazione S3 Express One Zone, ti consigliamo

di utilizzare gli SDK AWS per iniziare e gestire la richiesta API `CreateSession`. Per ulteriori informazioni sul modello basato sulla sessione, consulta Autorizzazione `CreateSession`.

Best practice per il checksum S3 aggiuntivo

S3 Express One Zone offre la possibilità di scegliere l'algoritmo di checksum utilizzato per convalidare i dati durante il caricamento o il download. Puoi selezionare uno dei seguenti algoritmi di controllo dell'integrità dei dati Secure Hash Algorithms (SHA) o Cyclic Redundancy Check (CRC): CRC32, CRC32C, SHA-1 e SHA-256. I checksum basati su MD5 non sono supportati con la classe di storage S3 Express One Zone.

CRC32 è il checksum predefinito utilizzato dagli SDK AWS durante la trasmissione di dati verso e da S3 Express One Zone. Ti consigliamo di utilizzare CRC32 e CRC32C per le migliori prestazioni con la classe di archiviazione S3 Express One Zone.

Utilizzo della versione più recente degli SDK AWS e delle librerie di runtime comuni

Diversi SDK AWS forniscono anche le librerie AWS Common Runtime (CRT) per accelerare ulteriormente le prestazioni nei client S3. Questi SDK includono AWS SDK for Java 2.x, AWS SDK for C++ e AWS SDK for Python (Boto3). Il client S3 basato su CRT trasferisce gli oggetti da e verso S3 Express One Zone con prestazioni e affidabilità migliorate utilizzando automaticamente l'operazione API di caricamento in più parti e i recuperi a intervallo di byte per automatizzare il dimensionamento orizzontale delle connessioni.

Per ottenere le migliori prestazioni con la classe di archiviazione S3 Express One Zone, si consiglia di utilizzare la versione più recente degli SDK AWS che include le librerie CRT o di utilizzare AWS Command Line Interface (AWS CLI).

Risoluzione dei problemi relativi alle prestazioni

Nuovi tentativi di richieste per applicazioni sensibili alla latenza

S3 Express One Zone è progettato appositamente per offrire livelli costanti di alte prestazioni senza ulteriori regolazioni. Tuttavia, l'impostazione di valori di timeout aggressivi e nuovi tentativi possono contribuire ulteriormente a garantire latenza e prestazioni costanti. Gli SDK AWS hanno valori di timeout e tentativo configurabili che puoi regolare in base alle tolleranze della tua applicazione.

Abbinamento di librerie AWS Common Runtime (CRT) e tipi di istanza Amazon EC2

Le applicazioni che eseguono un elevato numero di operazioni di lettura e scrittura richiedono una capacità di memoria o calcolo superiore rispetto alle applicazioni che non eseguono tali operazioni.

Durante il lancio delle istanze Amazon Elastic Compute Cloud (Amazon EC2) per carichi di lavoro esigenti in termini di prestazioni, scegli i tipi di istanza che dispongono della quantità di risorse necessaria per l'applicazione. L'archiviazione ad alte prestazioni S3 Express One Zone è abbinata idealmente a tipi di istanza più recenti con quantità maggiori di memoria di sistema, nonché CPU e GPU più potenti che possono sfruttare l'archiviazione ad alte prestazioni. È consigliabile inoltre utilizzare le versioni più recenti degli SDK AWS abilitati per CRT, che possono accelerare meglio le richieste di lettura e scrittura in parallelo.

Utilizzo dell'autenticazione basata sulla sessione negli SDK AWS anziché delle REST API HTTP

Con Amazon S3, puoi anche ottimizzare le prestazioni durante l'utilizzo delle richieste REST API HTTP seguendo le stesse best practice che fanno parte degli SDK AWS. Tuttavia, con il meccanismo di autorizzazione e autenticazione basato sulla sessione utilizzato da S3 Express One Zone, ti consigliamo di utilizzare gli SDK AWS per gestire `CreateSession` e il relativo token di sessione gestito. Gli SDK AWS creano e aggiornano automaticamente i token per tuo conto utilizzando l'operazione API `CreateSession`. L'uso di `CreateSession` consente di risparmiare sulla latenza di andata e ritorno per richiesta ad AWS Identity and Access Management (IAM) per autorizzare ciascuna richiesta.

Sviluppo con S3 Express One Zone

Amazon S3 Express One Zone è la prima classe di archiviazione S3 in cui è possibile selezionare una singola zona di disponibilità con la possibilità di co-ubicare l'archiviazione di oggetti con le risorse di calcolo, che offre la massima velocità di accesso possibile. Con la classe di archiviazione S3 Express One Zone, si utilizzano i bucket di directory S3 per archiviare i dati. Ogni bucket di directory utilizza la classe di archiviazione S3 Express One Zone per archiviare oggetti in una singola zona di disponibilità che è possibile selezionare al momento della creazione del bucket.

Dopo aver creato il bucket di directory, è possibile iniziare immediatamente letture e scritture a latenza molto bassa. Puoi comunicare con il bucket di directory mediante una connessione endpoint su un cloud privato virtuale (VPC) oppure puoi utilizzare le operazioni API zonali e regionali per gestire oggetti e bucket di directory. Puoi anche comunicare con la classe di archiviazione S3 Express One Zone tramite la console Amazon S3, la AWS Command Line Interface (AWS CLI), gli SDK AWS e la REST API Amazon S3.

La classe di storage Amazon S3 Express One Zone è progettata per una disponibilità del 99,95% all'interno di una singola zona di disponibilità ed è supportata dal Service Level Agreement di Amazon S3. Con S3 Express One Zone, i dati vengono archiviati in modo ridondante su più dispositivi

all'interno di una singola zona di disponibilità. S3 Express One Zone è progettata per gestire guasti simultanei dei dispositivi rilevando e riparando rapidamente l'eventuale ridondanza persa. Se il dispositivo esistente rileva un guasto, S3 Express One Zone sposta automaticamente le richieste in nuovi dispositivi all'interno di una zona di disponibilità. Questa ridondanza garantisce l'accesso ininterrotto ai dati all'interno di una zona di disponibilità.

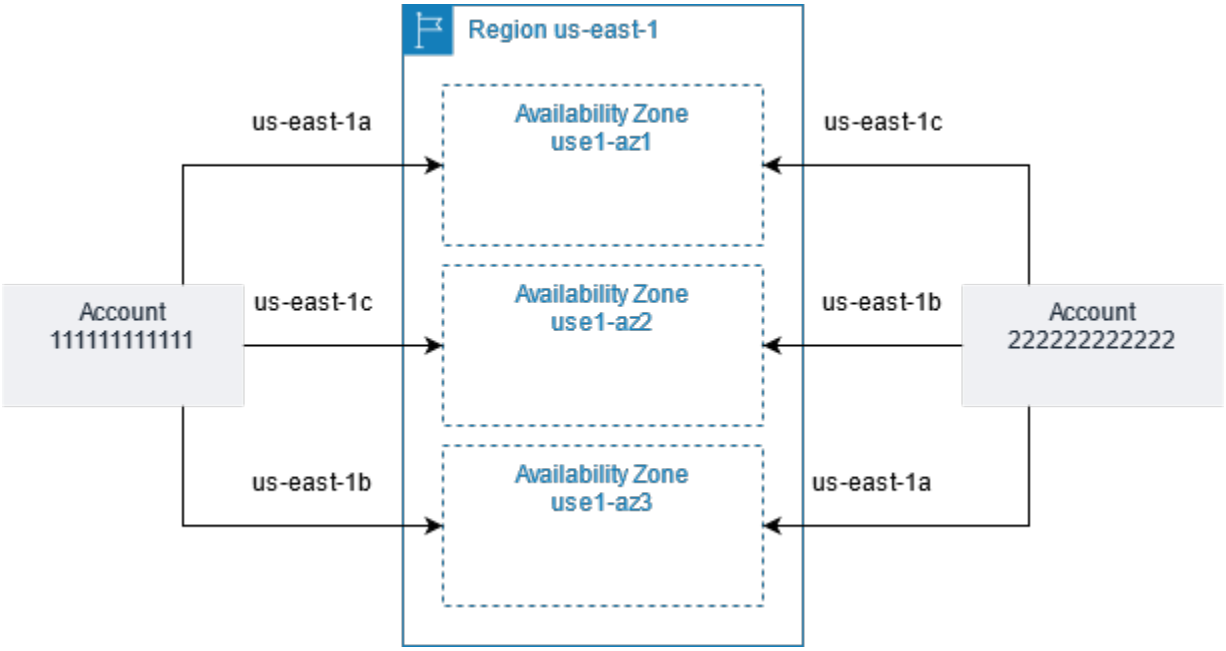
Argomenti

Zone di disponibilità e regioni S3 Express One Zone

Una zona di disponibilità consiste in uno o più data center separati con alimentazione, rete e connettività ridondanti in una Regione AWS. Per ottimizzare i recuperi a bassa latenza, gli oggetti della classe di archiviazione Amazon S3 Express One Zone vengono archiviati in modo ridondante in bucket di directory S3 in una singola zona di disponibilità che è locale per il carico di lavoro di calcolo. Quando crei un bucket di directory, scegli la zona di disponibilità e Regione AWS dove verrà posizionato il bucket.

AWS mappa le zone di disponibilità fisiche in modo casuale ai nomi delle zone di disponibilità di ciascuna. Account AWS Questo approccio consente di distribuire le risorse tra le zone di disponibilità in un' Regione AWS unica zona, anziché concentrarle probabilmente nella prima zona di disponibilità di ciascuna regione. Di conseguenza, la zona `us-east-1a` di disponibilità dell'utente Account AWS potrebbe non rappresentare la stessa posizione fisica `us-east-1a` di un'altra Account AWS. Per ulteriori informazioni, consulta Regioni e zone di disponibilità nella Guida per l'utente di Amazon EC2.

Per coordinare le zone di disponibilità tra account, devi utilizzare l'ID AZ, identificatore unico e coerente per una zona di disponibilità. Ad esempio, `use1-az1` è un ID AZ per la `us-east-1` regione e ha la stessa posizione fisica in ogni Account AWS regione. L'illustrazione seguente mostra come gli ID AZ siano gli stessi per ogni account, anche se i nomi delle zone di disponibilità potrebbero essere mappati in modo diverso per ciascun account.



Con S3 Express One Zone, i dati vengono archiviati in modo ridondante su più dispositivi all'interno di una singola zona di disponibilità. S3 Express One Zone è progettato per una disponibilità del 99,95% all'interno di una singola zona di disponibilità ed è supportato dal Service Level Agreement di Amazon S3. Per ulteriori informazioni, consulta Zona di disponibilità singola

S3 Express One Zone è supportato nelle seguenti regioni e zone di disponibilità:

Zone di disponibilità e regioni supportate da S3 Express One Zone

Nome Regione	Codice regione	ID zona di disponibilità
Stati Uniti orientali (Virginia settentrionale)	`us-east-1`	`use1-az4`
		`use1-az5`
		`use1-az6`
US West (Oregon)	`us-west-2`	`usw2-az1`
		`usw2-az3`
		`usw2-az4`

Nome Regione	Codice regione	ID zona di disponibilità
Asia Pacifico (Tokyo)	`ap-northeast-1`	`apne1-az1`
		`apne1-az4`
Europa (Stoccolma)	`eu-north-1`	`eun1-az1`
		`eun1-az2`
		`eun1-az3`

Endpoint regionali e zonali

Per accedere agli endpoint regionali e zonali per Amazon S3 Express One Zone dal cloud privato virtuale (VPC), puoi utilizzare endpoint VPC del gateway. Dopo aver creato un endpoint del gateway, puoi aggiungerlo come una destinazione nella tabella di routing per il traffico in transito dal VPC a S3 Express One Zone. L'utilizzo di endpoint gateway non comporta costi supplementari. Per ulteriori informazioni su come configurare gli endpoint VPC del gateway, consulta Servizi di rete per S3 Express One Zone.

Quando utilizzi S3 Express One Zone, le operazioni API a livello di bucket (o piano di controllo (control-plane)) sono disponibili tramite un endpoint regionale e sono denominate operazioni API degli endpoint regionali. Esempi di operazioni API degli endpoint regionali sono `CreateBucket` e `DeleteBucket`.

Dopo aver creato un bucket di directory, puoi utilizzare Zonal (operazioni API a livello di oggetto o data plane endpoint) per caricare e gestire gli oggetti nel tuo bucket di directory. Le operazioni API degli endpoint zonali sono disponibili tramite un endpoint zonale. Esempi di operazioni API zonali sono `PutObject` e `CopyObject`.

Operazioni API S3 Express One Zone

La classe di archiviazione Amazon S3 Express One Zone supporta operazioni API degli endpoint regionali (a livello di bucket o piano di controllo (control-plane)) e zonali (a livello di oggetto o piano dati). Per ulteriori informazioni, consultare Servizi di rete per S3 Express One Zone e Endpoint ed endpoint VPC del gateway.

Operazioni API degli endpoint regionali

Le seguenti operazioni API degli endpoint regionali sono supportate per S3 Express One Zone:

- CreateBucket
- DeleteBucket
- DeleteBucketPolicy
- GetBucketPolicy
- ListDirectoryBuckets
- PutBucketPolicy

Operazioni API degli endpoint zonali

Le seguenti operazioni API degli endpoint zonali sono supportate per S3 Express One Zone:

- CreateSession
- CopyObject
- DeleteObject
- DeleteObjects
- GetObject
- GetObjectAttributes
- HeadBucket
- HeadObject
- ListObjectsV2
- PutObject
- AbortMultipartUpload
- CompleteMultiPartUpload
- CreateMultipartUpload
- ListMultipartUploads
- ListParts
- UploadPart
- UploadPartCopy

Gestione dell'accesso ai dati con Punti di accesso Amazon S3

I punti di accesso Amazon S3 semplificano l'accesso ai dati per qualsiasi AWS servizio o applicazione del cliente che archivia dati in S3. Gli access point sono endpoint di rete denominati che vengono collegati a bucket che puoi usare per eseguire operazioni su oggetti S3, ad esempio `GetObject` e `PutObject`. Ogni access point dispone di autorizzazioni e controlli di rete distinti che S3 applica per qualsiasi richiesta effettuata tramite l'access point in questione. Ogni access point applica una policy di access point personalizzata che funziona in combinazione con la policy di bucket collegata al bucket sottostante. Puoi configurare qualsiasi access point per accettare le richieste solo da un cloud privato virtuale (VPC), in modo da limitare l'accesso ai dati Amazon S3 a una rete privata. È inoltre possibile configurare le impostazioni di blocco dell'accesso pubblico personalizzate per ciascun access point.

> **Note**
>
> - Puoi utilizzare gli access point solo per eseguire le operazioni sugli oggetti. Non puoi utilizzare access point per eseguire altre operazioni in Amazon S3, ad esempio la modifica o l'eliminazione dei bucket. Per un elenco completo delle operazioni S3 che supportano gli access point, consulta Compatibilità dei punti di accesso con i AWS servizi.
> - I punti di accesso funzionano con alcuni AWS servizi e funzionalità, ma non con tutti. Ad esempio, non è possibile configurare la replica tra regioni per operare tramite un access point. Per un elenco completo dei AWS servizi compatibili con i punti di accesso S3, consultaCompatibilità dei punti di accesso con i AWS servizi.

Questa sezione descrive come utilizzare Punti di accesso Amazon S3. Per informazioni sull'utilizzo di bucket, consulta Panoramica dei bucket. Per informazioni sull'utilizzo di oggetti, consulta Panoramica degli oggetti di Amazon S3.

Argomenti

Configurazione delle policy IAM per l'utilizzo degli access point

I punti di accesso Amazon S3 supportano politiche di risorse AWS Identity and Access Management (IAM) che consentono di controllare l'uso del punto di accesso in base alla risorsa, all'utente o ad altre condizioni. Affinché un'applicazione o un utente possa accedere agli oggetti tramite un access point, sia l'access point che il bucket sottostante devono consentire la richiesta.

> **Important**
>
> L'aggiunta di un punto di accesso S3 a un bucket non modifica il comportamento del bucket quando vi si accede direttamente tramite il nome del bucket o il nome della risorsa Amazon (ARN). Tutte le operazioni esistenti inerenti il bucket continueranno a funzionare come prima. Le limitazioni incluse in una policy di access point si applicano solo alle richieste effettuate tramite quell'access point.

Quando utilizzi le policy relative alle risorse IAM, assicurati di risolvere gli avvisi di sicurezza, gli errori, gli avvisi generali e i suggerimenti relativi alla sicurezza AWS Identity and Access Management Access Analyzer prima di salvare la policy. IAM Access Analyzer esegue controlli della policy per convalidarla in rapporto alla sintassi della policy e alle best practice di IAM. Questi controlli generano risultati e forniscono suggerimenti per aiutarti a creare policy funzionali e conformi alle best practice per la sicurezza.

Per ulteriori informazioni sulla convalida delle policy tramite IAM Access Analyzer, consulta Convalida delle policy di IAM Access Analyzer nella Guida per l'utente di IAM. Per visualizzare un elenco di avvisi, errori e suggerimenti di IAM Access Analyzer, consulta Riferimento ai controlli delle policy IAM Access Analyzer.

Esempi di policy degli access point

Negli esempi seguenti viene illustrato come creare le policy IAM per controllare le richieste effettuate tramite un access point.

> **Note**
>
> Le autorizzazioni concesse in una policy del punto di accesso sono valide solo se anche il bucket sottostante consente lo stesso accesso. Puoi farlo in due modi:

1. (Consigliato) Delega il controllo degli accessi dal bucket al punto di accesso come descritto in Delegazione del controllo di accesso agli access point.
2. Aggiungere le stesse autorizzazioni contenute nella policy del punto di accesso alla policy del bucket sottostante. Nel primo esempio di policy del punto di accesso viene illustrato come modificare la policy di bucket sottostante per consentire l'accesso necessario.

Example 1: Concessione della policy del punto di accesso

La policy del punto di accesso seguente concede all'utente IAM *Jane* dell'account *123456789012* le autorizzazioni per gli oggetti GET e PUT con il prefisso *Jane/* tramite il punto di accesso *my-access-point* nell'account *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

Affinché la policy di access point conceda effettivamente l'accesso ad *Jane*, anche il bucket sottostante deve consentire lo stesso accesso ad *Jane*. È possibile delegare il controllo di accesso dal bucket all'access point come descritto in Delegazione del controllo di accesso agli access point. In alternativa, è possibile aggiungere la policy seguente al bucket sottostante per concedere le autorizzazioni necessarie a Jane. Si noti che la voce `Resource` differisce tra le policy dell'access point e del 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:::DOC-EXAMPLE-BUCKET1/Jane/*"
    }]
}
```

Example 2: Policy del punto di accesso con condizione di tag

La policy del punto di accesso riportata di seguito concede all'utente IAM *Mateo* dell'account *123456789012* le autorizzazioni per gli oggetti GET tramite il punto di accesso *my-access-point* nell'account *123456789012* con la chiave di tag *data* impostata su un valore pari a *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: Policy del punto di accesso che consente l'elenco dei bucket

La policy del punto di accesso riportata di seguito consente all'utente IAM `Arnav` nell'account *123456789012* di visualizzare gli oggetti contenuti nel bucket sottostante il punto di accesso *my-access-point* nell'account *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: Policy di controllo dei servizi

La seguente policy di controllo dei servizi richiede la creazione di tutti i nuovi punti di accesso con un'origine di rete di tipo cloud privato virtuale (VPC). Con questa policy, gli utenti dell'organizzazione non possono creare nuovi access point accessibili da Internet.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Deny",
        "Action": "s3:CreateAccessPoint",
        "Resource": "*",
        "Condition": {
            "StringNotEquals": {
                "s3:AccessPointNetworkOrigin": "VPC"
            }
        }
    }]
}
```

Example 5: Policy di bucket per limitare le operazioni S3 alle origini di rete VPC

La policy di bucket seguente limita l'accesso a tutte le operazioni degli oggetti S3 per il bucket *example-s3-bucket* ai punti di accesso con un'origine di rete VPC.

> **Important**
>
> Prima di utilizzare un'istruzione come quella riportata nell'esempio, assicurati di non utilizzare funzionalità non supportate dai punti di accesso, come la replica tra regioni.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:BypassGovernanceRetention",
                "s3:DeleteObject",
                "s3:DeleteObjectTagging",
                "s3:DeleteObjectVersion",
                "s3:DeleteObjectVersionTagging",
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectLegalHold",
                "s3:GetObjectRetention",
                "s3:GetObjectTagging",
                "s3:GetObjectVersion",
                "s3:GetObjectVersionAcl",
                "s3:GetObjectVersionTagging",
                "s3:ListMultipartUploadParts",
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:PutObjectLegalHold",
                "s3:PutObjectRetention",
                "s3:PutObjectTagging",
                "s3:PutObjectVersionAcl",
                "s3:PutObjectVersionTagging",
                "s3:RestoreObject"
            ],
            "Resource": "arn:aws:s3:::example-s3-bucket/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:AccessPointNetworkOrigin": "VPC"
                }
```

```
            }
        }
    ]
}
```

Chiavi di condizione

I punti di accesso S3 utilizzano chiavi di condizione che possono essere utilizzate nelle policy IAM per controllare l'accesso alle risorse. Le seguenti chiavi di condizione rappresentano solo una parte di una policy IAM. Per esempi completi di policy, consulta Esempi di policy degli access point, the section called “Delegazione del controllo di accesso agli access point” e the section called “Concessione delle autorizzazioni per i punti di accesso multi-account”.

`s3:DataAccessPointArn`

Questo esempio mostra una stringa che è possibile utilizzare per la corrispondenza dell'ARN di un punto di accesso. L'esempio seguente corrisponde a tutti i punti di accesso per Account AWS *123456789012* in Region: *us-west-2*

```
"Condition" : {
    "StringLike": {
        "s3:DataAccessPointArn": "arn:aws:s3:us-west-2:123456789012:accesspoint/*"
    }
}
```

`s3:DataAccessPointAccount`

Questo esempio mostra un operatore stringa che è possibile utilizzare per la corrispondenza dell'ID account del proprietario di un punto di accesso. L'esempio seguente restituisce tutti i punti di accesso di proprietà dell' Account AWS *123456789012*.

```
"Condition" : {
    "StringEquals": {
        "s3:DataAccessPointAccount": "123456789012"
    }
}
```

`s3:AccessPointNetworkOrigin`

Questo esempio mostra un operatore stringa che è possibile utilizzare per la corrispondenza dell'origine di rete, `Internet` o `VPC`. L'esempio seguente esegue la corrispondenza solo degli access point con un'origine VPC.

```
"Condition" : {
    "StringEquals": {
        "s3:AccessPointNetworkOrigin": "VPC"
    }
}
```

Per ulteriori informazioni sull'utilizzo delle chiavi di condizione con Amazon S3, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.

Delegazione del controllo di accesso agli access point

È possibile delegare il controllo degli accessi per un bucket agli access point del bucket. La policy di bucket di esempio seguente consente l'accesso completo a tutti i punti di accesso dell'account del proprietario del bucket. Pertanto, tutto l'accesso a questo bucket è controllato dalle policy associate agli access point. Si consiglia di configurare i bucket in questo modo per tutti i casi d'uso che non richiedono l'accesso diretto al bucket.

Example 6: Policy di bucket che delega il controllo degli accessi ai punti di accesso

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS": "*" },
        "Action" : "*",
        "Resource" : [ "Bucket ARN", "Bucket ARN/*"],
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointAccount" : "Bucket owner's account
 ID" }
        }
    }]
}
```

Concessione delle autorizzazioni per i punti di accesso multi-account

Per creare un punto di accesso a un bucket di proprietà di un altro account, devi prima creare il punto di accesso specificando il nome del bucket e l'ID del proprietario dell'account. Il proprietario del bucket deve quindi aggiornare la policy di bucket per autorizzare le richieste dal punto di accesso. La creazione di un punto di accesso è simile alla creazione di un DNS CNAME in quanto il punto di accesso non fornisce l'accesso al contenuto del bucket. Tutti gli accessi ai bucket sono controllati dalla policy di bucket. La policy di bucket di esempio consente di eseguire richieste `GET` e `LIST` sul bucket da un punto di accesso di proprietà di un Account AWS attendibile.

Sostituisci *`Bucket ARN`* con l'ARN del bucket.

Example 7 — Policy di Bucket che delega le autorizzazioni a un altro Account 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" }
        }
    }]
}
```

Creazione di access point

Amazon S3 fornisce le funzionalità per la creazione e la gestione degli access point. Puoi creare punti di accesso S3 utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDK o l'API REST di Amazon S3.

Per impostazione predefinita, puoi creare fino a 10.000 punti di accesso per regione per ciascuno dei tuoi Account AWS. Se hai bisogno di più di 10.000 punti di accesso per un singolo account in una singola regione, puoi richiedere un aumento della quota di servizio. Per ulteriori informazioni su Service Quotas e la richiesta di un aumento, consultare AWS Service Quotas in Riferimenti generali di AWS.

> **Note**
>
> Poiché potresti voler pubblicizzare il nome del punto di accesso per consentire ad altri utenti di utilizzarlo, ti consigliamo di evitare di includere informazioni sensibili nel nome del punto di accesso. I nomi dei punti di accesso vengono pubblicati in un database accessibile pubblicamente noto come sistema dei nomi di dominio (DNS).

Regole per la denominazione degli Punti di accesso Amazon S3

I nomi degli access point devono soddisfare le condizioni seguenti:

- Deve essere univoco all'interno di una singola regione Account AWS
- Devono rispettare le limitazioni di denominazione DNS
- Devono iniziare con un numero o una lettera minuscola
- Devono contenere da 3 a 50 caratteri
- Non possono iniziare o terminare con un trattino (`-`).
- Non possono contenere caratteri di sottolineatura (`_`), lettere maiuscole o punti (`.`).
- Impossibile terminare con il suffisso `-s3alias`. Questo suffisso è riservato ai nomi alias dei punti di accesso. Per ulteriori informazioni, consulta Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3.

Per creare un punto di accesso, consulta i seguenti argomenti.

Argomenti

- Creazione di un access point
- Creazione di access point limitati a un cloud privato virtuale
- Gestione dell'accesso pubblico agli access point

Creazione di un access point

Un punto di accesso è associato esattamente a un bucket Amazon S3. Se desideri utilizzare un bucket nel tuo Account AWS, devi prima crearne uno. Per ulteriori informazioni sulla creazione dei bucket, consulta Creazione, configurazione e utilizzo di bucket Amazon S3.

Puoi anche creare un punto di accesso multi-account associato a un bucket in un altro Account AWS, purché tu conosca il nome del bucket e l'ID dell'account del proprietario del bucket. Tuttavia, la creazione di punti di accesso multi-account non consente l'accesso ai dati nel bucket finché non vengono concesse le autorizzazioni dal proprietario del bucket. Il proprietario del bucket deve concedere all'account del proprietario del punto di accesso (il tuo account) l'accesso al bucket tramite la policy di bucket. Per ulteriori informazioni, consulta Concessione delle autorizzazioni per i punti di accesso multi-account.

Per impostazione predefinita, puoi creare fino a 10.000 punti di accesso per regione per ciascuno dei tuoi Account AWS. Se hai bisogno di più di 10.000 punti di accesso per un singolo account in una singola regione, puoi richiedere un aumento della quota di servizio. Per ulteriori informazioni su Service Quotas e la richiesta di un aumento, consultare AWS Service Quotas in Riferimenti generali di AWS.

Gli esempi seguenti mostrano come creare un punto di accesso con la console AWS CLI e S3. Per ulteriori informazioni su come creare punti di accesso tramite REST API, consulta CreateAccessPoint nella Documentazione di riferimento delle API Amazon Simple Storage Service.

Utilizzo della console S3

Per creare un punto di accesso

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nella barra di navigazione nella parte superiore della pagina, scegli il nome del file attualmente visualizzato Regione AWS. Quindi, scegli la regione in cui desideri creare un punto di accesso.
3. Nel pannello di navigazione a sinistra, scegli Access Points (Punti di accesso).
4. Nella pagina Access Points (Punti di accesso) scegli Create access point (Crea punto di accesso).
5. Nel campo Nome del punto di accesso immetti il nome per il punto di accesso. Per ulteriori informazioni sui punti di accesso S3, consulta Regole per la denominazione degli Punti di accesso Amazon S3.
6. Per Nome bucket specifica il bucket S3 che desideri utilizzare con il punto di accesso.

   Per usare un bucket nel tuo account, seleziona Scegli un bucket in questo account e digita o cerca il nome del bucket.

Per utilizzare un bucket in un altro account Account AWS, scegli Specificare un bucket in un altro account e inserisci l' Account AWS ID e il nome del bucket.

> **Note**
>
> Se utilizzi un bucket in un altro Account AWS, il proprietario del bucket deve aggiornare la policy del bucket per autorizzare le richieste dal punto di accesso. Per un esempio di policy di bucket, consulta Concessione delle autorizzazioni per i punti di accesso multi-account.

7. Scegli una Origine della rete. Se si sceglie Virtual Private Cloud (VPC), immettere l'ID VPC che si desidera utilizzare con il punto di accesso.

   Per ulteriori informazioni sulle origini di rete per i punti di accesso, consulta Creazione di access point limitati a un cloud privato virtuale.

8. In Block Public Access settings for this Access Point (Impostazioni punto di accesso per blocco dell'accesso pubblico), seleziona le impostazioni di blocco dell'accesso pubblico da applicare all'access point. Tutte le impostazioni di blocco dell'accesso pubblico sono abilitate per impostazione predefinita per i nuovi punti di accesso. È consigliabile lasciare tutte le impostazioni abilitate, a meno che tu non debba necessariamente disabilitarne una specifica.

   > **Note**
   >
   > Dopo aver creato un punto di accesso, non è più possibile modificare le impostazioni di blocco dell'accesso pubblico.

   Per ulteriori informazioni sull'uso del blocco dell'accesso pubblico di Amazon S3 con i punti di accesso, consulta Gestione dell'accesso pubblico agli access point.

9. (Facoltativo) In Policy del punto di accesso - facoltativo, specificare la policy dell'access point. Prima di salvare la policy, assicurati di risolvere gli avvisi di sicurezza, gli errori, gli avvisi generali e i suggerimenti. Per ulteriori informazioni sulla specifica di una policy dei punti di accesso, consulta Esempi di policy degli access point.

10. Selezionare Crea punto di accesso.

Usando il AWS CLI

Il comando di esempio seguente crea un punto di accesso denominato *example-ap* per il bucket *DOC-EXAMPLE-BUCKET* nell'account *111122223333*. Per creare il punto di accesso, devi inviare una richiesta ad Amazon S3 che specifica quanto segue:

- Nome del punto di accesso. Per informazioni sulle regole di denominazione, consulta the section called “Regole per la denominazione degli Punti di accesso Amazon S3”.
- Nome del bucket a cui si desidera associare il punto di accesso.
- L'ID dell'account del Account AWS proprietario del bucket.

```
aws s3control create-access-point --name example-ap --account-id 111122223333 --
bucket DOC-EXAMPLE-BUCKET
```

Quando crei un punto di accesso utilizzando un bucket in un altro Account AWS, includi il `--bucket-account-id` parametro. Il seguente comando di esempio crea un punto di accesso nell' Account AWS *111122223333*, utilizzando il bucket *DOC-EXAMPLE-BUCKET2*, che si trova nell' Account AWS *444455556666*.

```
aws s3control create-access-point --name example-ap --account-id 111122223333 --
bucket DOC-EXAMPLE-BUCKET --bucket-account-id 444455556666
```

Creazione di access point limitati a un cloud privato virtuale

Quando crei un access point, puoi scegliere di rendere l'access point accessibile da Internet oppure specificare che tutte le richieste effettuate attraverso l'access point devono provenire da un cloud privato virtuale (VPC) specifico. Un access point accessibile da Internet ha l'origine di rete `Internet`. Può essere utilizzato da qualsiasi punto di Internet, fatte salve altre limitazioni di accesso in vigore per l'access point, il bucket sottostante e le risorse correlate, come gli oggetti richiesti. Un access point accessibile solo da un VPC specificato ha l'origine di rete VPC e Amazon S3 rifiuta qualsiasi richiesta fatta all'access point che non provenga da quel VPC.

> **Important**
>
> Puoi specificare l'origine di rete di un access point solo quando crei l'access point. Dopo aver creato l'access point, non è più possibile modificare l'origine di rete.

Per limitare un access point all'accesso solo VPC, è necessario includere il parametro `VpcConfiguration` con la richiesta di creare l'access point. Nel parametro `VpcConfiguration`, specificare l'ID VPC che si desidera utilizzare l'access point. Se una richiesta viene effettuata tramite il punto di accesso, la richiesta deve provenire dal VPC o Amazon S3 la rifiuterà.

Puoi recuperare l'origine di rete di un punto di accesso utilizzando gli AWS SDK o AWS CLI le API REST. Se per un access point è specificata una configurazione VPC, la sua origine di rete è `VPC`. In caso contrario, l'origine della rete dell'access point è `Internet`.

Example

Esempio: creazione di un punto di accesso limitato all'accesso VPC

Nell'esempio seguente viene creato un punto di accesso denominato `example-vpc-ap` per il bucket `example-bucket` nell'account `123456789012` che consente l'accesso solo dal VPC `vpc-1a2b3c`. L'esempio verifica quindi che il nuovo access point abbia l'origine di rete `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"
}
```

Per utilizzare un access point con un VPC, è necessario modificare la policy di accesso per l'endpoint VPC. Gli endpoint VPC consentono al traffico di fluire dal VPC ad Amazon S3. Dispongono di policy di controllo dell'accesso che controllano il modo in cui le risorse all'interno del VPC possono interagire con Amazon S3. Le richieste dal VPC ad Amazon S3 hanno esito positivo solo tramite un punto di accesso se la policy dell'endpoint VPC concede l'accesso sia al punto di accesso che al bucket sottostante.

> **Note**
>
> Per rendere le risorse accessibili solo all'interno di un VPC, assicurati di creare una zona ospitata privata per l'endpoint VPC. Per utilizzare una zona ospitata privata, modificare le impostazioni del VPC in modo che gli attributi di rete VPC `enableDnsHostnames` e `enableDnsSupport` siano impostati su `true`.

L'istruzione di policy di esempio riportata di seguito configura un endpoint VPC per consentire le chiamate a `GetObject` per un bucket denominato `awsexamplebucket1` e un access point denominato `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 dichiarazione `"Resource"` in questo esempio utilizza un Amazon Resource Name (ARN) per specificare l'access point. Per ulteriori informazioni sugli ARN dei punti di accesso, consulta la sezione Utilizzo degli access point.

Per ulteriori informazioni sulle policy degli endpoint VPC, consulta Utilizzo delle policy degli endpoint per Amazon S3 nella Guida per l'utente del VPC.

Gestione dell'accesso pubblico agli access point

Gli Punti di accesso Amazon S3 supportano le impostazioni di blocco dell'accesso pubblico indipendenti per ciascun access point. Quando crei un access point, puoi specificare le impostazioni di blocco dell'accesso pubblico applicabili all'access point. Per qualsiasi richiesta effettuata tramite un access point, Amazon S3 valuta le impostazioni di blocco dell'accesso pubblico per l'access point, il bucket sottostante e l'account del proprietario del bucket. Se una di queste impostazioni indica che la richiesta deve essere bloccata, Amazon S3 rifiuta la richiesta.

Per ulteriori informazioni sulla funzionalità di blocco dell'accesso pubblico S3, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

> **Important**
>
> - Tutte le impostazioni di blocco dell'accesso pubblico sono abilitate per impostazione predefinita per gli access point. È necessario disabilitare esplicitamente le impostazioni che non vuoi applicare a un access point.
> - Amazon S3 attualmente non supporta la modifica delle impostazioni di blocco dell'accesso pubblico di un punto di accesso dopo la creazione del punto di accesso.

Example

Esempio: creare un access point con impostazioni di blocco dell'accesso pubblico personalizzate

In questo esempio viene creato un access point denominato `example-ap` per il bucket `example-bucket` nell'account `123456789012` con impostazioni di blocco dell'accesso pubblico non

predefinite. L'esempio recupera quindi la configurazione del nuovo access point per verificarne le impostazioni di blocco dell'accesso pubblico.

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"
}
```

Utilizzo degli access point

Puoi accedere agli oggetti in un bucket Amazon S3 con un punto di accesso utilizzando gli AWS Management Console, AWS CLI, AWS SDK o le API REST di S3.

Gli access point hanno l'Amazon Resource Name (ARN). Gli ARN dei punti di accesso sono simili agli ARN dei bucket ma vengono digitati in modo esplicito e codificano la regione del punto di accesso e l'ID Account AWS del proprietario del punto di accesso. Per ulteriori informazioni sull'utilizzo degli ARN, consulta Amazon Resource Name (ARN) nella Riferimenti generali di AWS.

Gli ARN di access point utilizzano il formato `arn:aws:s3:`*`region`*`:`*`account-id`*`:accesspoint/`*`resource`*. Ad esempio:

- arn:aws:s3:us-west-2:123456789012:accesspoint/test rappresenta l'access point denominato `test`, di proprietà dell'account `123456789012` nella regione `us-west-2`.

- arn:aws:s3:us-west-2:123456789012:accesspoint/* rappresenta tutti gli access point dell'account `123456789012` nella regione `us-west-2`.

Gli ARN per gli oggetti a cui si accede tramite un access point utilizzano il formato `arn:aws:s3:region:account-id:accesspoint/access-point-name/object/resource`. Ad esempio:

- arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/unit-01 rappresenta l'oggetto `unit-01`, accessibile tramite l'access point denominato `test`, di proprietà dell'account `123456789012` nella regione `us-west-2`.
- arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/* rappresenta tutti gli oggetti per l'access point `test`, dell'account `123456789012` nella regione `us-west-2`.
- arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/unit-01/finance/* rappresenta tutti gli oggetti del prefisso `unit-01/finance/` per l'access point `test`, dell'account `123456789012` nella regione `us-west-2`.

Accesso a un bucket tramite gli Access Point S3

Gli access point S3 supportano solo l'indirizzamento. virtual-host-style Per indirizzare un bucket tramite un punto di accesso, utilizza il formato seguente.

```
https://AccessPointName-AccountId.s3-accesspoint.region.amazonaws.com
```

> **Note**
>
> - Se il nome del punto di accesso include trattini (-), includere i trattini nell'URL e inserire un altro trattino prima dell'ID account. Ad esempio, per utilizzare un punto di accesso denominato `finance-docs` di proprietà dell'account `123456789012` nella regione `us-west-2`, l'URL appropriato è `https://finance-docs-123456789012.s3-accesspoint.us-west-2.amazonaws.com`.
> - I punti di accesso S3 non supportano l'accesso tramite HTTP, ma solo l'accesso protetto da HTTPS.

Argomenti

- Monitoraggio e registrazione degli access point
- Gestione e utilizzo degli Punti di accesso Amazon S3 nella console di Amazon S3
- Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3
- Utilizzo di punti di accesso con operazioni compatibili con Amazon S3

Se disponi di un cloud privato virtuale (VPC), consultare Gestione dell'accesso ad Amazon S3 con endpoint VPC e punti di accesso S3.

Monitoraggio e registrazione degli access point

Amazon S3 registra le richieste effettuate tramite i punti di accesso e le richieste effettuate alle API che gestiscono i punti di accesso, ad esempio `CreateAccessPoint` e `GetAccessPointPolicy`. Per monitorare e gestire i modelli di utilizzo, puoi anche configurare i parametri delle richieste di Amazon CloudWatch Logs per i punti di accesso.

Argomenti

- CloudWatch parametri di richiesta
- Registri delle richieste

CloudWatch parametri di richiesta

Per comprendere e migliorare le prestazioni delle applicazioni che utilizzano punti di accesso, puoi utilizzare CloudWatch i parametri di richiesta di Amazon S3. I parametri di richiesta ti aiutano a monitorare le richieste di Amazon S3 per identificare rapidamente i problemi operativi e intraprendere le operazioni appropriate.

Per impostazione predefinita, questi parametri sono disponibili a livello di bucket. Tuttavia, puoi definire un filtro per i parametri di richiesta utilizzando un prefisso condiviso, tag oggetto o un punto di accesso. Quando crei un filtro con un punto di accesso, la configurazione dei parametri della richiesta include le richieste al punto di accesso specificato. Puoi ricevere parametri, impostare allarmi e accedere ai pannelli di controllo per visualizzare le operazioni eseguite in tempo reale tramite questo punto di accesso.

Devi acconsentire esplicitamente ai parametri di richiesta configurandoli nella console o utilizzando l'API Amazon S3. I parametri di richiesta sono disponibili a intervalli di 1 minuto dopo una determinata latenza per l'elaborazione. I parametri delle richieste vengono fatturati alla stessa tariffa dei parametri personalizzati. CloudWatch Per ulteriori informazioni, consulta i CloudWatch prezzi di Amazon.

Per creare una configurazione dei parametri di richiesta che filtra in base al punto di accesso, vedi Creazione di una configurazione dei parametri che filtra in base al prefisso, al tag oggetto o al punto di accesso.

Registri delle richieste

Puoi registrare le richieste effettuate tramite i punti di accesso e le richieste effettuate alle API che li gestiscono, ad esempio `CreateAccessPoint` e `GetAccessPointPolicy`, utilizzando la registrazione degli accessi al server e AWS CloudTrail.

CloudTrail le voci di registro per le richieste effettuate tramite punti di accesso includono l'ARN del punto di accesso nella `resources` sezione del registro.

Si prenda come esempio la seguente configurazione:

- Un bucket denominato `DOC-EXAMPLE-BUCKET1` nella regione `us-west-2` che contiene un oggetto denominato `my-image.jpg`
- Un access point denominato `my-bucket-ap` associato a `DOC-EXAMPLE-BUCKET1`
- Un Account AWS ID di `123456789012`

L'esempio seguente mostra la `resources` sezione di una voce di CloudTrail registro per la configurazione precedente:

```
"resources": [
        {"type": "AWS::S3::Object",
            "ARN": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/my-image.jpg"
        },
        {"accountId": "123456789012",
            "type": "AWS::S3::Bucket",
            "ARN": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
        },
        {"accountId": "123456789012",
            "type": "AWS::S3::AccessPoint",
            "ARN": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-bucket-ap"
        }
    ]
```

Per ulteriori informazioni sui log degli accessi al server S3, consulta Registrazione delle richieste con registrazione dell'accesso al server. Per ulteriori informazioni su AWS CloudTrail, vedere What is AWS CloudTrail? nella Guida AWS CloudTrail per l'utente.

Gestione e utilizzo degli Punti di accesso Amazon S3 nella console di Amazon S3

Questa sezione illustra come gestire e utilizzare gli Punti di accesso Amazon S3 utilizzando la AWS Management Console. Prima di iniziare, passa alla pagina dei dettagli del punto di accesso che desideri gestire o utilizzare, come descritto nella procedura seguente.

Argomenti

Visualizzazione degli access point per il tuo account

Per elencare tutti i punti di accesso creati nel Account AWS

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nella barra di navigazione nella parte superiore della pagina, scegli il nome del file attualmente visualizzato Regione AWS. Quindi, scegli la regione per cui desideri elencare i punti di accesso.
3. Nel riquadro di navigazione sul lato sinistro della console scegliere Punti di accesso.
4. Nella pagina dei punti di accesso, sotto Punti di accesso, visualizza i punti di accesso del tuo Regione AWS.
5. (Facoltativo) cerca i punti di accesso in base al nome immettendo un nome nel campo di testo accanto al menu a discesa delle regioni.
6. Scegliere il nome del punto di accesso che si desidera gestire o utilizzare.

Visualizzazione degli access point per un bucket

Per elencare tutti i punti di accesso presenti in te Account AWS per un singolo bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nella barra di navigazione nella parte superiore della pagina, scegli il nome del file attualmente visualizzato, Regione AWS quindi scegli la regione per cui desideri elencare i punti di accesso.
3. Nel riquadro di navigazione sul lato sinistro della console scegliere Bucket.
4. Nella pagina Buckets (Bucket) seleziona il nome del bucket di cui si desidera elencare i punti di accesso.
5. Nella pagina dei dettagli del bucket scegliere la scheda Access points (Punti di accesso).
6. Scegliere il nome del punto di accesso che si desidera gestire o utilizzare.

Per visualizzare i dettagli di configurazione per un access point

1. Passare alla pagina dei dettagli del punto di accesso di cui si desidera visualizzare i dettagli, come descritto in Visualizzazione degli access point per il tuo account.
2. In Access point overview (Panoramica dei punti di accesso), visualizza i dettagli e le proprietà della configurazione per il punto di accesso selezionato.

Eliminazione di un access point

1. Passare alla pagina dei dettagli del punto di accesso per il punto di accesso che si desidera utilizzare, come descritto in Visualizzazione degli access point per il tuo account.
2. Nella scheda Objects (Oggetti) scegliere il nome di uno o più oggetti a cui si desidera accedere tramite il punto di accesso. Nella console viene visualizzata un'etichetta sopra il nome del bucket che mostra il punto di accesso attualmente in uso. Durante l'utilizzo dell'access point, è possibile eseguire solo le operazioni sugli oggetti consentite dalle autorizzazioni dell'access point.

> **Note**
>
> - La visualizzazione della console mostra sempre tutti gli oggetti presenti nel bucket. L'utilizzo di un access point tipo quello descritto in questa procedura limita le

operazioni che puoi eseguire sugli oggetti, ma non la visualizzazione degli oggetti presenti nel bucket.

- La console di gestione S3 non supporta l'utilizzo di access point del cloud privato virtuale (VPC, Virtual Private Cloud) per accedere alle risorse bucket. Per accedere alle risorse del bucket da un punto di accesso VPC, usa AWS CLI gli SDK o le AWS API REST di Amazon S3.

Visualizzazione delle impostazioni di blocco dell'accesso pubblico per un access point

1. Passare alla pagina dei dettagli del punto di accesso di cui si desidera visualizzare le impostazioni, come descritto in Visualizzazione degli access point per il tuo account.
2. Seleziona Autorizzazioni.
3. In Access point policy (Policy del punto di accesso) esamina le impostazioni per il blocco dell'accesso pubblico del punto di accesso.

> **Note**
>
> Non è possibile modificare le impostazioni del blocco dell'accesso pubblico per un punto di accesso dopo la sua creazione.

Modifica di una policy dell'access point

1. Passare alla pagina dei dettagli del punto di accesso di cui si desidera modificare la policy, come descritto in Visualizzazione degli access point per il tuo account.
2. Seleziona Autorizzazioni.
3. In Access point policy (Policy del punto di accesso) scegliere Edit (Modifica).
4. Immettere la policy del punto di accesso nel campo di testo. La console visualizza automaticamente l'ARN (Amazon Resource Name del punto di accesso che può essere utilizzato nella policy.

Eliminazione di un punto di accesso

1. Passa all'elenco dei punti di accesso per l'account o per un bucket specifico, come descritto in Visualizzazione degli access point per il tuo account.

2. Seleziona il pulsante di opzione accanto al nome dell'access point che si desidera eliminare.
3. Scegliere Delete (Elimina).
4. Confermare di voler eliminare il punto di accesso inserendone il nome nel campo di testo che viene visualizzato, quindi scegliere Confirm (Conferma).

Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3

Quando crei un punto di accesso, Amazon S3 genera automaticamente un alias che puoi utilizzare al posto di un nome bucket Amazon S3 per l'accesso ai dati. Puoi utilizzare questo alias del punto di accesso al posto di un nome della risorsa Amazon (ARN) per qualsiasi operazione del piano dati del punto di accesso. Per un elenco di queste operazioni, consulta Compatibilità dei punti di accesso con i AWS servizi.

Di seguito viene illustrato un esempio di ARN e alias del punto di accesso per un punto di accesso denominato *my-access-point*.

- ARN: `arn:aws:s3:`*region*`:`*account-id*`:accesspoint/`*my-access-point*
- Alias del punto di accesso: *my-access-point*`-`*hrzrlukc5m36ft7okagglf3gmwluquse1b*`-s3alias`

Per ulteriori informazioni sull'utilizzo degli ARN, consulta Amazon Resource Name (ARN) nella Riferimenti generali di AWS.

Nomi degli alias del punto di accesso

Un nome alias punto di accesso viene creato nello stesso spazio dei nomi di un bucket Amazon S3. Questo nome alias viene generato automaticamente e non può essere modificato. Un nome alias del punto di accesso soddisfa tutti i requisiti di un nome bucket Amazon S3 valido e comprende le seguenti parti:

access point prefix`-`*metadata*`-s3alias`

> **Note**
>
> Il suffisso `-s3alias` è riservato ai nomi alias dei punti di accesso e non può essere utilizzato per i nomi dei bucket o dei punti di accesso. Per ulteriori informazioni sulle regole di denominazione dei bucket Amazon S3, consulta Regole di denominazione dei bucket.

Casi d'uso e limitazioni degli alias dei punti di accesso

Quando si adottano i punti di accesso, è possibile utilizzare nomi alias dei punti di accesso senza richiedere modifiche estese di codice.

Quando crei un punto di accesso, Amazon S3 genera automaticamente un nome alias del punto di accesso, come mostrato nell'esempio seguente. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
aws s3control create-access-point --bucket example-s3-bucket1 --name my-access-point --
account-id 111122223333
{
    "AccessPointArn":
    "arn:aws:s3:region:111122223333:accesspoint/my-access-point",
    "Alias": "my-access-point-aqfqprnstn7aefdfbarligizwgyfouse1a-s3alias"
}
```

Puoi utilizzare questo nome alias del punto di accesso invece di un nome bucket Amazon S3 in qualsiasi operazione del piano dati. Per un elenco di queste operazioni, consulta Compatibilità dei punti di accesso con i AWS servizi.

L' AWS CLI esempio seguente del `get-object` comando utilizza l'alias del punto di accesso del bucket per restituire informazioni sull'oggetto specificato. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
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": {}
}
```

Limitazioni

- Gli alias non possono essere configurati dai clienti.

- Gli alias non possono essere eliminati, modificati o disabilitati in un punto di accesso.
- È possibile utilizzare questo nome di alias del punto di accesso al posto di un nome di bucket Amazon S3 in alcune operazioni del piano dati. Per un elenco di queste operazioni, consulta Compatibilità dei punti di accesso con le operazioni S3.
- Non puoi utilizzare un nome alias del punto di accesso per le operazioni del piano di controllo Amazon S3. Per un elenco delle operazioni del piano di controllo Amazon S3, consulta Controllo Amazon S3 nellaDocumentazione di riferimento delle API di Amazon Simple Storage Service.
- Non puoi utilizzare gli alias dei punti di accesso S3 come origine o destinazione per le operazioni Move nella console Amazon S3.
- Gli alias non possono essere utilizzati nelle policy AWS Identity and Access Management (IAM).
- Gli alias non possono essere utilizzati come destinazione di registrazione per i log di accesso al server S3.
- Gli alias non possono essere utilizzati come destinazione di registrazione per i log. AWS CloudTrail
- Amazon SageMaker GroundTruth non supporta gli alias dei punti di accesso.

Utilizzo di punti di accesso con operazioni compatibili con Amazon S3

Negli esempi seguenti viene illustrato come utilizzare i punti di accesso con operazioni compatibili in Amazon S3.

Argomenti

- Compatibilità dei punti di accesso con i AWS servizi
- Compatibilità dei punti di accesso con le operazioni S3
- Richiedere un oggetto tramite un punto di accesso
- Caricamento di un oggetto tramite un alias del punto di accesso
- Eliminare un oggetto tramite un punto di accesso
- Visualizzazione di oggetti tramite un alias del punto di accesso
- Aggiungere un set di tag a un oggetto tramite un punto di accesso
- Concedere autorizzazioni di accesso tramite un punto di accesso utilizzando un'ACL

Compatibilità dei punti di accesso con i AWS servizi

Gli alias dei punti di accesso Amazon S3 consentono a qualsiasi applicazione che richiede un nome bucket S3 di utilizzare facilmente un punto di accesso. È possibile utilizzare gli alias dei punti di

accesso S3 ovunque si utilizzino nomi bucket S3 per accedere ai dati in S3. Per ulteriori informazioni, consulta Casi d'uso e limitazioni degli alias dei punti di accesso.

Compatibilità dei punti di accesso con le operazioni S3

Puoi utilizzare i punti di accesso per accedere a un bucket utilizzando il seguente sottoinsieme di API Amazon S3. Tutte le operazioni elencate di seguito possono accettare ARN o alias dei punti di accesso:

Operazioni S3

- AbortMultipartUpload
- CompleteMultipartUpload
- CopyObject (solo copie nella stessa regione)
- 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 copie nella stessa regione)

Richiedere un oggetto tramite un punto di accesso

L'esempio seguente mostra come richiedere l'oggetto `my-image.jpg` tramite l'access point `prod` di proprietà dell'ID account `123456789012` nella regione `us-west-2` e salvare il file scaricato come `download.jpg`.

AWS CLI

```
aws s3api get-object --key my-image.jpg --bucket arn:aws:s3:us-
west-2:123456789012:accesspoint/prod download.jpg
```

Caricamento di un oggetto tramite un alias del punto di accesso

Nell'esempio seguente l'oggetto `my-image.jpg` viene caricato tramite l'alias del punto di accesso `my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias` di proprietà dell'ID account `123456789012` nella regione `us-west-2`.

AWS CLI

```
aws s3api put-object --bucket my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-
s3alias --key my-image.jpg --body my-image.jpg
```

Eliminare un oggetto tramite un punto di accesso

Nell'esempio seguente l'oggetto `my-image.jpg` viene eliminato tramite il punto di accesso `prod` di proprietà dell'ID account `123456789012` nella regione `us-west-2`.

AWS CLI

```
aws s3api delete-object --bucket arn:aws:s3:us-west-2:123456789012:accesspoint/prod
 --key my-image.jpg
```

Visualizzazione di oggetti tramite un alias del punto di accesso

Nell'esempio seguente vengono elencati gli oggetti tramite l'alias del punto di accesso `my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias` di proprietà dell'ID account `123456789012` nella regione `us-west-2`.

AWS CLI

```
aws s3api list-objects-v2 --bucket my-access-point-
hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias
```

Aggiungere un set di tag a un oggetto tramite un punto di accesso

L'esempio seguente aggiunge un set di tag all'oggetto esistente `my-image.jpg` tramite l'access point `prod` di proprietà dell'ID account `123456789012` nella regione `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"}]
```

Concedere autorizzazioni di accesso tramite un punto di accesso utilizzando un'ACL

L'esempio seguente applica una ACL a un oggetto esistente `my-image.jpg` tramite l'access point `prod` di proprietà dell'ID account `123456789012` nella regione `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
```

Restrizioni e limitazioni degli access point.

Gli Punti di accesso Amazon S3 presentano le seguenti restrizioni e limitazioni:

- Ogni access point è associato esattamente a un bucket che deve essere specificato quando crei l'access point. Dopo aver creato un access point, non è possibile associarlo a un bucket diverso. Tuttavia, puoi eliminare un punto di accesso e quindi crearne un altro con lo stesso nome e associarlo a un bucket diverso.
- I nomi dei punti di accesso devono soddisfare determinate condizioni. Per ulteriori informazioni sui punti di accesso S3, consulta Regole per la denominazione degli Punti di accesso Amazon S3.
- Dopo aver creato un access point, non è possibile modificarne la configurazione del cloud privato virtuale (VPC).
- Le policy access point sono limitate a una dimensione di 20 KB.
- È possibile creare un massimo di 10.000 punti di accesso Account AWS per regione. Se hai bisogno di più di 10.000 punti di accesso per un singolo account in una singola regione, puoi richiedere un aumento della quota di servizio. Per ulteriori informazioni su Service Quotas e la richiesta di un aumento, consultare AWS Service Quotas in Riferimenti generali di AWS.
- Regioni AWS Se disponi di più di 1.000 punti di accesso, non puoi cercare un punto di accesso per nome nella console Amazon S3.
- Non è possibile utilizzare un punto di accesso come destinazione della replica S3. Per ulteriori informazioni sulla replica, consulta Panoramica sulla replica degli oggetti.
- Non puoi utilizzare gli alias dei punti di accesso S3 come origine o destinazione per le operazioni Move nella console Amazon S3.
- Puoi indirizzare i punti di accesso solo utilizzando gli URL. virtual-host-style Per ulteriori informazioni sull' virtual-host-style indirizzamento, vedereAccesso ed elenco di un bucket Amazon S3.
- Le operazioni API che controllano la funzionalità dei punti di accesso (ad esempio, `PutAccessPoint` e `GetAccessPointPolicy`) non supportano le chiamate multi-account.

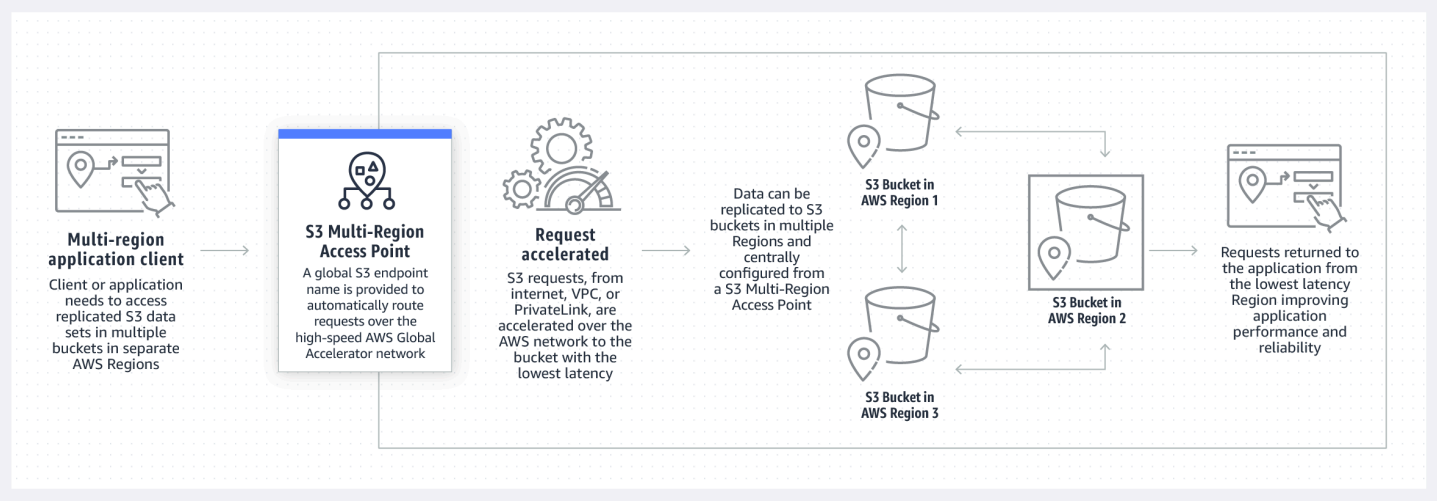
- È necessario utilizzare AWS Signature Version 4 quando si effettuano richieste a un punto di accesso utilizzando le API REST. Per ulteriori informazioni sull'autenticazione delle richieste, consulta Authenticating Requests (AWS Signature Version 4) nel riferimento all'API di Amazon Simple Storage Service.
- Gli access point supportano solo le richieste tramite HTTPS. Amazon S3 risponderà automaticamente con un reindirizzamento HTTP a tutte le richieste effettuate tramite HTTP, per aggiornare la richiesta a HTTPS.
- Gli access point non supportano l'accesso anonimo.
- I punti di accesso multi-account non concedono l'accesso ai dati finché non ti vengono concesse le autorizzazioni dal proprietario del bucket. Il proprietario del bucket mantiene sempre il massimo controllo sui dati e deve aggiornare la policy di bucket per autorizzare le richieste provenienti dal punto di accesso multi-account. Per un esempio di policy di bucket, consulta Configurazione delle policy IAM per l'utilizzo degli access point.
- Quando visualizzi un punto di accesso multi-account nella console Amazon S3, la colonna Accesso mostra Sconosciuto. La console Amazon S3 non è in grado di determinare se l'accesso pubblico è concesso per il bucket e gli oggetti associati. A meno che non sia necessaria una configurazione pubblica per un caso d'uso specifico, consigliamo all'utente e al proprietario del bucket di bloccare tutti gli accessi pubblici al punto di accesso e al bucket. Per ulteriori informazioni, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

Punti di accesso multi-regione in Amazon S3

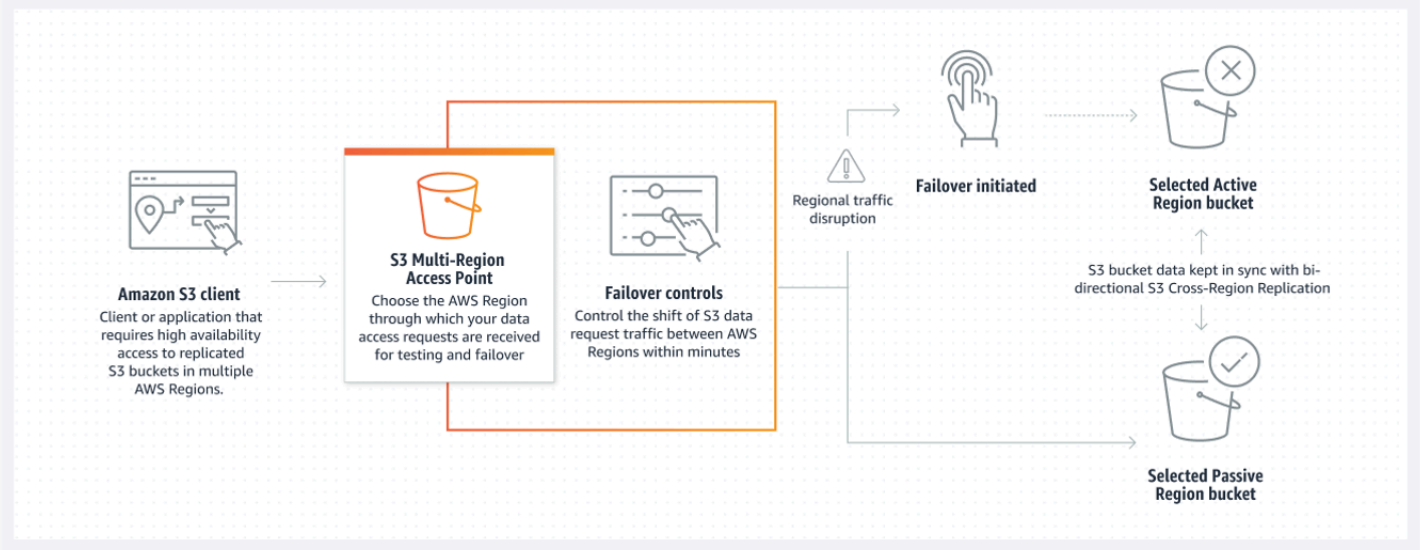
I punti di accesso multi-regione di Amazon S3 forniscono un endpoint globale che le applicazioni possono utilizzare per eseguire le richieste provenienti da bucket S3 situati in più Regioni AWS. Puoi utilizzare i punti di accesso multi-regione per creare applicazioni multi-regione con la stessa architettura utilizzata in una singola regione e quindi eseguire tali applicazioni in qualsiasi parte del mondo. Invece di inviare richieste sulla rete Internet pubblica e congestionata, i punti di accesso multi-regione offrono la resilienza di rete integrata con l'accelerazione delle richieste basate su Internet ad Amazon S3. Le richieste di applicazioni effettuate a un endpoint globale Multi-Region Access Point vengono utilizzate AWS Global Acceleratorper instradare automaticamente attraverso la rete AWS globale verso il bucket S3 più vicino con uno stato di routing attivo.

Quando si crea un punto di accesso multiregionale, si specifica un set di Regioni AWS dove si desidera archiviare i dati da fornire tramite tale punto di accesso multiregionale. Puoi utilizzare la Replica tra regioni S3 per sincronizzare i dati tra i bucket in tali regioni. Puoi quindi richiedere o scrivere dati tramite l'endpoint globale del punto di accesso multi-regione. Amazon S3 gestisce automaticamente le richieste al set di dati replicato dalla regione disponibile più vicina. I punti di accesso multi-regione sono inoltre compatibili con le applicazioni in esecuzione nei cloud privati virtuali (VPC) Amazon, incluse quelle che utilizzano AWS PrivateLink per Amazon S3.

L'immagine seguente è una rappresentazione grafica di un punto di accesso multi-regione Amazon S3 in una configurazione multi-regione. Il grafico mostra come le richieste Amazon S3 vengono indirizzate automaticamente ai bucket nella Regione AWS attiva più vicina.



L'immagine seguente è una rappresentazione grafica di un punto di accesso multi-regione Amazon S3 in una configurazione attiva-passiva. Il grafico illustra come controllare il traffico di accesso ai dati di Amazon S3 per passare tra Regioni AWS attive e passive.



Per ulteriori informazioni su come utilizzare i punti di accesso multi-regione, consulta la sezione Tutorial: Nozioni di base sui punti di accesso multi-regione di Amazon S3.

Argomenti

- Creazione di punti di accesso multi-regione
- Configurazione di un punto di accesso multi-regione per l'utilizzo con AWS PrivateLink
- Esecuzione di richieste utilizzando un punto di accesso multi-regione

Creazione di punti di accesso multi-regione

Per creare un punto di accesso multi-regione in Amazon S3, esegui le operazioni seguenti:

- Specifica il nome del punto di accesso multi-regione.
- Scegli un bucket in ognuno Regione AWS dei quali desideri soddisfare le richieste per il punto di accesso multiregionale.
- Configura le impostazioni per il blocco dell'accesso pubblico di Amazon S3 per il punto di accesso multi-regione.

Fornisci tutte queste informazioni in una richiesta di creazione, che Amazon S3 elabora in modo asincrono. Amazon S3 offre un token che consente di monitorare lo stato della richiesta di creazione asincrona.

Assicurati di risolvere avvisi di sicurezza, errori, avvisi generali e suggerimenti da AWS Identity and Access Management Access Analyzer prima di salvare la policy. IAM Access Analyzer esegue controlli della policy per convalidarla in rapporto alla sintassi della policy e alle best practice di IAM. Questi controlli generano risultati e forniscono suggerimenti utili per aiutarti a creare policy funzionali e conformi alle best practice per la sicurezza. Per ulteriori informazioni sulla convalida delle policy tramite IAM Access Analyzer, consulta IAM Access Analyzer policy validation (Convalida delle policy di IAM Access Analyzer) nella Guida per l'utente di IAM. Per visualizzare un elenco di avvisi, errori e suggerimenti di IAM Access Analyzer, consulta Riferimento ai controlli delle policy IAM Access Analyzer.

Quando utilizzi l'API, la richiesta di creare un punto di accesso multi-regione è asincrona. Quando invii una richiesta di creazione di un punto di accesso multi-regione, Amazon S3 autorizza la richiesta in modo sincrono. Quindi restituisce immediatamente un token che consente di monitorare lo stato di avanzamento della richiesta di creazione. Per ulteriori informazioni sulla registrazione delle richieste asincrone per creare e gestire punti di accesso multi-regione, consulta Utilizzo dei punti di accesso multi-regione con operazioni API supportate.

Dopo aver creato il punto di accesso multi-regione, puoi creare per esso una policy di controllo degli accessi. Ogni punto di accesso multi-regione può avere una policy associata. Le policy dei punti di accesso multi-regione sono policy basate su risorse che consentono di limitare l'utilizzo del punto di accesso multi-regione per risorsa, utente o altre condizioni.

> **Note**
>
> Affinché un'applicazione o un utente possa accedere a un oggetto tramite un punto di accesso multi-regione, entrambe le policy seguenti devono consentire la richiesta:
>
> - La policy di accesso per il punto di accesso multi-regione.
> - La policy di accesso per il bucket sottostante contenente l'oggetto
>
> Quando le due policy sono diverse, ha la precedenza quella più restrittiva.

Per semplificare la gestione delle autorizzazioni per i punti di accesso multi-regione, puoi delegare il controllo degli accessi dal bucket al punto di accesso multi-regione. Per ulteriori informazioni, consulta the section called “Esempi di policy dei punti di accesso multi-regione”.

L'utilizzo di un bucket con un punto di accesso multi-regione non modifica il comportamento di un bucket a cui si accede tramite il nome del bucket esistente o un nome della risorsa Amazon (ARN). Tutte le operazioni esistenti inerenti il bucket continuano a funzionare come prima. Le limitazioni incluse in una policy per un punto di accesso multi-regione si applicano solo alle richieste effettuate tramite quell'access point multi-regione.

Dopo aver creato la policy per un punto di accesso multi-regione, puoi aggiornarla ma non puoi eliminarla. Puoi tuttavia aggiornare la policy del punto di accesso multi-regione in modo che neghi tutte le autorizzazioni.

Argomenti

Regole per la denominazione dei punti di accesso multi-regione in Amazon S3

Quando crei un punto di accesso multi-regione, gli assegni un nome, ovvero una stringa scelta da te. Dopo la creazione, non puoi modificare il nome del punto di accesso multi-regione. Il nome deve essere univoco nel tuo Account AWS e deve essere conforme ai requisiti di denominazione elencati in Restrizioni e limitazioni dei punti di accesso multi-regione. Per facilitare l'identificazione del punto di accesso multi-regione, utilizza un nome significativo per te o per l'organizzazione oppure che rispecchi lo scenario.

Utilizzerai questo nome per richiamare le operazioni di gestione di un punto di accesso multi-regione, ad esempio `GetMultiRegionAccessPoint` e `PutMultiRegionAccessPointPolicy`.

Il nome non viene utilizzato per inviare richieste al punto di accesso multi-regione e non deve necessariamente essere esposto ai client che effettuano richieste utilizzando il punto di accesso multi-regione.

Quando Amazon S3 crea un punto di accesso multi-regione, gli assegna automaticamente un alias. Questo alias è una stringa alfanumerica univoca che termina in `.mrap`. L'alias viene utilizzato per costruire il nome host e l'ARN (Amazon Resource Name) per un punto di accesso multi-regione. Il nome completo si basa anche sull'alias del punto di accesso multi-regione.

Non è possibile determinare il nome di un punto di accesso multi-regione dal relativo alias, pertanto puoi divulgare un alias senza il rischio di esporre il nome, lo scopo o il proprietario del punto di accesso multi-regione. Amazon S3 seleziona l'alias per ogni nuovo punto di accesso multi-regione e l'alias non può essere modificato. Per ulteriori informazioni sull'indirizzamento di un access point multi-regione, consultaEsecuzione di richieste utilizzando un punto di accesso multi-regione.

Gli alias del punto di accesso multi-regione sono univoci nel tempo e non si basano sul suo nome né sulla sua configurazione. Se crei un punto di accesso multi-regione, quindi lo elimini e ne crei un altro con lo stesso nome e la stessa configurazione, il secondo punto di accesso multi-regione avrà un alias diverso dal primo. I nuovi punti di accesso multi-regione non possono mai avere lo stesso alias di uno precedente.

Regole per la scelta dei bucket per i punti di accesso multi-regione in Amazon S3

Ogni punto di accesso multi-regione è associato alle regioni in cui desideri evadere le richieste. Il punto di accesso multi-regione deve essere associato esattamente a un bucket in ciascuna di queste regioni. Specifica il nome di ogni bucket nella richiesta per creare il punto di accesso multi-regione. I bucket che supportano il punto di accesso multiregionale possono trovarsi nello stesso Account AWS che possiede il punto di accesso multiregionale oppure in un altro. Account AWS

Un singolo bucket può essere utilizzato da più punti di accesso multi-regione.

> **Important**
>
> - Puoi specificare i bucket associati a un punto di accesso multi-regione solo al momento della creazione. Dopo la creazione, non puoi aggiungere, modificare o rimuovere i bucket dalla configurazione del punto di accesso multi-regione. Per modificare i bucket, devi eliminare l'intero punto di accesso multi-regione e crearne uno nuovo.

- Non puoi eliminare un bucket che fa parte di un punto di accesso multi-regione. Se desideri eliminare un bucket associato a un punto di accesso multi-regione, elimina prima il punto di accesso multi-regione.
- Se al punto di accesso multi-regione aggiungi un bucket di proprietà di un altro account, il proprietario del bucket deve aggiornare anche la policy di bucket per concedere le autorizzazioni di accesso al punto di accesso multi-regione. In caso contrario, il punto di accesso multi-regione non sarà in grado di recuperare i dati dal bucket. Per alcune policy di esempio che illustrano come concedere tale accesso, consulta Esempi di policy dei punti di accesso multi-regione.
- Non tutte le regioni supportano i punti di accesso multiregione. Per vedere l'elenco delle regioni supportate, consulta Restrizioni e limitazioni dei punti di accesso multi-regione.

Puoi creare regole di replica per sincronizzare i dati tra i bucket. Queste regole consentono di copiare automaticamente i dati dai bucket di origine ai bucket di destinazione. La connessione di bucket a un punto di accesso multi-regione non influisce sul funzionamento della replica. La configurazione della replica con punti di accesso multi-regione viene descritta in una sezione successiva.

Important

Quando esegui una richiesta su un punto di accesso multi-regione, tale punto di accesso non è a conoscenza del contenuto dei dati dei bucket nel punto di accesso multi-regione. Pertanto, il bucket che riceve la richiesta potrebbe non contenere i dati richiesti. Per creare set di dati coerenti nei bucket Amazon S3 associati a un punto di accesso multi-regione, ti consigliamo di configurare la replica tra regioni di S3 (CRR). Per ulteriori informazioni, consulta Configurazione della replica per l'utilizzo con punti di accesso multi-regione.

Creare un punto di accesso multi-regione in Amazon S3

Negli esempi seguenti viene illustrato come creare un punto di accesso multi-regione utilizzando la console Amazon S3.

Utilizzo della console S3

Per creare un punto di accesso multi-regione

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Multi-Region Access Points (Punti di accesso multi-regione).
3. Scegli Crea punti di accesso multi-regione per iniziare a creare il punto di accesso multi-regione.
4. Nella pagina Punto di accesso multi-regione, specifica un nome per il punto di accesso multi-regione nel campo Nome del punto di accesso multi-regione.
5. Seleziona i bucket che verranno associati a questo punto di accesso multi-regione. Puoi scegliere i bucket che si trovano nel tuo account oppure puoi scegliere i bucket da altri account.

> **Note**
>
> Devi aggiungere almeno un bucket dal tuo account o da altri account. Inoltre, tieni presente che i punti di accesso multi-regione supportano un solo bucket per ogni Regione AWS. Pertanto, non puoi aggiungere due bucket dalla stessa regione. Non sono supportate le Regioni AWS disattivate per impostazione predefinita.

- Per aggiungere un bucket presente nel tuo account, scegli Aggiungi bucket. Viene visualizzato un elenco dei bucket disponibili nel tuo account. Puoi cercare il tuo bucket per nome o ordinare i nomi dei bucket in ordine alfabetico.
- Per aggiungere un bucket da un altro account, scegli Aggiungi bucket da altri account. Assicurati di conoscere il nome e l' Account AWS ID esatti del bucket, perché non puoi cercare o cercare i bucket in altri account.

> **Note**
>
> Devi inserire un Account AWS ID e un nome di bucket validi. Il bucket deve inoltre trovarsi in una regione supportata; in caso contrario, si verificherà un errore quando tenti di creare il punto di accesso multi-regione. Per l'elenco delle regioni che supportano i punti di accesso multi-regione, consulta Restrizioni e limitazioni dei punti di accesso multi-regione.

6. (Facoltativo) Se devi rimuovere un bucket aggiunto, scegli Rimuovi.

> **Note**
>
> Non puoi aggiungere o rimuovere bucket a questo punto di accesso multi-regione dopo averlo creato.

7. In Block Public Access settings for this Multi-Region Access Point (Impostazioni di blocco dell'accesso pubblico per il punto di accesso multi-regione), seleziona le impostazioni di blocco dell'accesso pubblico da applicare al punto di accesso. Per impostazione predefinita, tutte le impostazioni di blocco dell'accesso pubblico sono abilitate per i nuovi punti di accesso multi-regione. È consigliabile lasciare tutte le impostazioni abilitate, a meno che tu non debba necessariamente disabilitarne una specifica.

> **Note**
>
> Non è possibile modificare le impostazioni del blocco dell'accesso pubblico per un punto di accesso multi-regione dopo la sua creazione. Pertanto, se intendi bloccare l'accesso pubblico, assicurati che le tue applicazioni funzionino correttamente senza accesso pubblico prima di creare un punto di accesso multi-regione.

8. Scegli Create Multi-Region Access Point (Crea punto di accesso multi-regione).

> **Important**
>
> Se al punto di accesso multi-regione aggiungi un bucket di proprietà di un altro account, il proprietario del bucket deve aggiornare anche la policy di bucket per concedere le autorizzazioni di accesso al punto di accesso multi-regione. In caso contrario, il punto di accesso multi-regione non sarà in grado di recuperare i dati dal bucket. Per alcune policy di esempio che illustrano come concedere tale accesso, consulta Esempi di policy dei punti di accesso multi-regione.

Usando il AWS CLI

È possibile utilizzare il AWS CLI per creare un punto di accesso multiregionale. Quando crei il punto di accesso multi-regione, devi specificare tutti i bucket che supporterà. Non è possibile aggiungere bucket al punto di accesso multi-regione dopo che il punto è stato creato.

Nell'esempio seguente viene creato un punto di accesso multi-regione con due bucket utilizzando la AWS CLI. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
aws s3control create-multi-region-access-point --account-id 111122223333 --details '{
        "Name": "simple-multiregionaccesspoint-with-two-regions",
        "PublicAccessBlock": {
            "BlockPublicAcls": true,
            "IgnorePublicAcls": true,
            "BlockPublicPolicy": true,
            "RestrictPublicBuckets": true
        },
        "Regions": [
            { "Bucket": "example-s3-bucket1" },
            { "Bucket": "example-s3-bucket2" }
        ]
    }' --region us-west-2
```

Blocco dell'accesso pubblico con i punti di accesso multi-regione di Amazon S3

Ogni punto di accesso multi-regione dispone di impostazioni distinte per il blocco dell'accesso pubblico di Amazon S3. Queste impostazioni funzionano insieme alle impostazioni di blocco dell'accesso pubblico per il proprietario del Account AWS punto di accesso multiregionale e dei bucket sottostanti.

Quando Amazon S3 autorizza una richiesta, applica la combinazione più restrittiva di queste impostazioni. Se le impostazioni di blocco dell'accesso pubblico per una di queste risorse (l'account proprietario del punto di accesso multi-regione, il bucket sottostante o l'account proprietario del bucket) bloccano l'accesso per l'azione o la risorsa richiesta, Amazon S3 rifiuta la richiesta.

È consigliabile abilitare tutte le impostazioni di blocco dell'accesso pubblico a meno che non sia necessario disabilitarne alcune. Per impostazione predefinita, tutte le impostazioni di blocco dell'accesso pubblico sono abilitate per i punti di accesso multi-regione. Se il blocco dell'accesso pubblico è abilitato, il punto di accesso multi-regione non è in grado di accettare richieste basate su Internet.

> **Important**
>
> Dopo la creazione del punto di accesso multi-regione, non puoi più modificare le relative impostazioni di blocco dell'accesso pubblico.

Per ulteriori informazioni sul blocco dell'accesso pubblico in Amazon S3, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

Visualizzazione dei dettagli della configurazione dei punti di accesso multi-regione S3

Nell'esempio seguente viene illustrato come visualizzare i dettagli di configurazione del punto di accesso multi-regione utilizzando la console Amazon S3.

Utilizzo della console S3

Per creare un punto di accesso multi-regione

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Multi-Region Access Points (Punti di accesso multi-regione).
3. Scegli il nome del punto di accesso multi-regione di cui si desidera visualizzare la configurazione.

   - Nella scheda Proprietà sono elencati tutti i bucket associati al punto di accesso multi-regione, la data di creazione, il nome della risorsa Amazon (ARN) e l'alias. Nella colonna ID Account AWS sono riportati anche tutti i bucket di proprietà di account esterni associati al punto di accesso multi-regione.
   - Nella scheda Autorizzazioni sono elencate le impostazioni di blocco dell'accesso pubblico applicate ai bucket associati a questo punto di accesso multi-regione. Puoi anche visualizzare la policy del punto di accesso multi-regione per il tuo punto di accesso multi-regione, se ne hai creato uno. L'avviso Informazioni nella pagina Autorizzazioni include anche tutti i bucket (nel tuo account e in altri account) per questo punto di accesso multi-regione con l'impostazione L'accesso pubblico è bloccato abilitata.
   - La scheda Replica e failover fornisce una visualizzazione in formato mappa dei bucket associati al punto di accesso multi-regione e delle regioni in cui risiedono i bucket. Se sono presenti bucket di un altro account per i quali non disponi delle autorizzazioni per estrarrne i

dati, la regione verrà contrassegnata in rosso sulla mappa Riepilogo della replica, a indicare che si tratta di una Regione AWS che ha generato errori durante il recupero dello stato della replica.

> **Note**
>
> Per recuperare le informazioni sullo stato della replica da un bucket in un account esterno, il proprietario del bucket deve concederti l'autorizzazione `s3:GetBucketReplication`nella propria policy dei bucket.

Questa scheda fornisce anche le metriche di replica, le regole di replica e gli stati di failover per le regioni utilizzate con il punto di accesso multi-regione.

Usando il AWS CLI

È possibile utilizzare il AWS CLI per visualizzare i dettagli di configurazione per un punto di accesso multiregionale.

L' AWS CLI esempio seguente ottiene la configurazione corrente del punto di accesso multiregionale. Per utilizzare questo comando di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

```
aws s3control get-multi-region-access-point --account-id 111122223333 --name example-s3-bucket1
```

Eliminazione di un punto di accesso multi-regione

La procedura seguente spiega come eliminare un punto di accesso multi-regione utilizzando la console Amazon S3.

L'eliminazione di un punto di accesso multi-regione non comporta l'eliminazione dei bucket associati al punto di accesso multi-regione, ma solo del punto di accesso multi-regione stesso.

Utilizzo della console S3

Per creare un punto di accesso multi-regione

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

2. Nel pannello di navigazione a sinistra, scegli Multi-Region Access Points (Punti di accesso multi-regione).
3. Seleziona il pulsante di opzione accanto al nome del punto di accesso multi-regione.
4. Scegli Elimina.
5. Nella finestra di dialogo Elimina punto di accesso multiregionale, inserisci il nome del AWS bucket che desideri eliminare.

> **Note**
>
> Assicurati di inserire un nome valido per il bucket. In caso contrario, il pulsante Elimina verrà disabilitato.

6. Scegli Elimina per confermare l'eliminazione del punto di accesso multi-regione.

Usando il AWS CLI

È possibile utilizzare il AWS CLI per eliminare un punto di accesso multiregionale. Questa operazione non elimina i bucket associati al punto di accesso multi-regione, ma solo al punto di accesso multi-regione stesso. Per utilizzare questo comando di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

```
aws s3control delete-multi-region-access-point --account-id 123456789012 --details
 Name=example-multi-region-access-point-name
```

Configurazione di un punto di accesso multi-regione per l'utilizzo con AWS PrivateLink

Puoi utilizzare i punti di accesso multi-regione per instradare il traffico delle richieste Amazon S3 tra Regioni AWS. Ogni endpoint globale del punto di accesso multi-regione instrada il traffico delle richieste di dati Amazon S3 da più fonti senza dover creare configurazioni di rete complesse con endpoint distinti. Queste origini del traffico delle richieste di dati includono:

- Traffico con origine in un cloud privato virtuale (VPC)
- Traffico proveniente da data center on-premise e gestito da AWS PrivateLink
- Traffico proveniente dalla rete Internet pubblica

Se stabilisci una connessione AWS PrivateLink a un punto di accesso multi-regione S3, puoi instradare le richieste S3 ad AWS o a più Regioni AWS su una connessione privata utilizzando un'architettura e una configurazione di rete semplici. Quando si utilizza AWS PrivateLink, non è necessario configurare una connessione peering VPC.

Argomenti

Configurazione di un punto di accesso multi-regione per l'utilizzo con AWS PrivateLink

AWS PrivateLink offre la connettività privata ad Amazon S3 utilizzando indirizzi IP privati nel virtual private cloud (VPC). Puoi effettuare il provisioning di uno o più endpoint di interfaccia all'interno del tuo VPC per connetterti ai punti di accesso multi-regione di Amazon S3.

Puoi creare endpoint com.amazonaws.s3-global.accesspointper punti di accesso multi-regione tramite la AWS Management Console, AWS CLI, oppure gli SDK AWS. Per ulteriori informazioni su come configurare un endpoint di interfaccia per i punti di accesso multi-regione, consulta Endpoint VPC dell'interfaccia nella Guida dell'utente di VPC.

Per effettuare richieste a un punto di accesso multi-regione tramite endpoint di interfaccia, segui la procedura riportata di seguito per configurare il VPC e il punto di accesso multi-regione.

Per configurare un punto di accesso multi-regione da utilizzare con AWS PrivateLink

1. Crea o disponi di un endpoint VPC appropriato in grado di connettersi a punti di accesso multi-regione. Per ulteriori informazioni sulla creazione di endpoint VPC, consulta Endpoint VPC di interfaccia nella Guida per l'utente di VPC.

   > **Important**
   >
   > Assicurati di creare un endpoint com.amazonaws.s3-global.accesspoint. Altri tipi di endpoint non possono accedere ai punti di accesso multi-regione.

Dopo aver creato questo endpoint VPC, tutte le richieste del punto di accesso multi-regione nel VPC si instradano attraverso questo endpoint se hai abilitato il DNS privato per l'endpoint. Questo è abilitato per impostazione predefinita.

2. Se la policy del punto di accesso multi-regione non supporta le connessioni dagli endpoint VPC, dovrai aggiornarlo.
3. Verifica che le policy dei singoli bucket consentano l'accesso agli utenti del punto di accesso multi-regione.

Ricorda che i punti di accesso multi-regione funzionano instradando le richieste ai bucket, non soddisfacendo le richieste stesse. È importante ricordare che l'origine della richiesta deve disporre delle autorizzazioni per il punto di accesso multi-regione e deve poter accedere ai singoli bucket del punto di accesso multi-regione. In caso contrario, la richiesta potrebbe essere instradata a un bucket in cui l'origine non dispone delle autorizzazioni per soddisfare la richiesta. Un punto di accesso multi-regione e i bucket associati possono essere di proprietà dello stesso account AWS o di un altro account. Tuttavia, i VPC di account diversi possono utilizzare un punto di accesso multi-regione se le autorizzazioni sono configurate correttamente.

Per questo motivo, la policy dell'endpoint VPC deve consentire l'accesso sia al punto di accesso multi-regione che a ogni bucket sottostante che desideri che sia in grado di soddisfare le richieste. Ad esempio, supponiamo di avere un punto di accesso multi-regione con l'alias `mfzwi23gnjvgw.mrap`. È supportato dai bucket `DOC-EXAMPLE-BUCKET1` e `DOC-EXAMPLE-BUCKET2`, tutti di proprietà dell'account AWS `123456789012`. In questo caso, le seguenti policy dell'endpoint VPC consente ai bucket di supporto di soddisfare le richieste `GetObject` dal VPC fatte a `mfzwi23gnjvgw.mrap`.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Sid": "Read-buckets-and-MRAP-VPCE-policy",
        "Principal": "*",
        "Action": [
            "s3:GetObject"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*",
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET2/*",
```

```
            "arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/*"
        ]
    }]
}
```

Come accennato in precedenza, devi inoltre assicurarti che la policu del punto di accesso multi-regione sia configurata in modo da supportare l'accesso tramite un endpoint VPC. Non è necessario specificare l'endpoint VPC che richiede l'accesso. La policy di esempio seguente concede l'accesso a qualsiasi richiedente che tenta di utilizzare il punto di accesso multi-regione per le richieste `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/*"
    }]
}
```

E, naturalmente, i singoli bucket avrebbero bisogno di una policy per supportare l'accesso delle richieste inviate tramite l'endpoint VPC. La policy di esempio seguente consente l'accesso in lettura a tutti gli utenti anonimi, incluse le richieste effettuate tramite l'endpoint VPC.

```
{
    "Version":"2012-10-17",
   "Statement": [
   {
       "Sid": "Public-read",
       "Effect":"Allow",
       "Principal": "*",
       "Action": "s3:GetObject",
       "Resource": [
           "arn:aws:s3:::DOC-EXAMPLE-BUCKET1",
           "arn:aws:s3:::DOC-EXAMPLE-BUCKET2/*"]
    }]
```

```
}
```

Per informazioni sulla modifica di una policy dell'endpoint VPC, consulta Controllare l'accesso ai servizi con endpoint VPC nella Guida per l'utente di VPC.

Rimozione dell'accesso a un punto di accesso multi-regione da un endpoint VPC

Se sei proprietario di un punto di accesso multi-regione e desideri rimuovere l'accesso a tale punto da un endpoint di interfaccia, devi specificare una nuova policy di accesso per il punto di accesso multi-regione che impedisca l'accesso alle richieste provenienti dagli endpoint VPC. Tuttavia, se i bucket nel punto di accesso multi-regione supportano le richieste tramite endpoint VPC, essi continueranno a supportarle. Se desideri impedire il supporto, devi aggiornare anche le policy dei bucket. L'impostazione di una nuova policy di accesso al punto di accesso multi-regione impedisce l'accesso solo a tale punto di accesso e non ai bucket sottostanti.

> **Note**
>
> Non puoi eliminare una policy di accesso per un punto di accesso multi-area. Per rimuovere l'accesso a un punto di accesso multi-area, devi fornire una nuova policy di accesso con l'accesso modificato come desideri.

Invece di aggiornare la policy di accesso per il punto di accesso multi-regione, puoi aggiornare le policy di bucket per impedire le richieste tramite gli endpoint VPC. In questo caso, gli utenti potrebbero comunque accedere al punto di accesso multi-regione tramite l'endpoint VPC. Tuttavia, se viene instradata a un bucket la cui policy impedisce l'accesso, la richiesta al punto di accesso multi-regione genera un messaggio di errore.

Esecuzione di richieste utilizzando un punto di accesso multi-regione

Come altre risorse, i punti di accesso multi-regione di Amazon S3 sono simile ai nomi delle risorse Amazon (ARN). È possibile utilizzare questi ARN per indirizzare le richieste ai punti di accesso multi-regione utilizzando AWS Command Line Interface (AWS CLI), gli SDK AWS oppure l'API Amazon S3. È inoltre possibile utilizzare questi ARN per identificare i punti di accesso multi-regione nelle

policy di controllo degli accessi. L'ARN di un punto di accesso multi-regione non include né rivela il nome del punto di accesso corrispondente. Per ulteriori informazioni sull'utilizzo degli ARN, consulta Amazon Resource Name (ARN) nella Riferimenti generali di AWS.

> **Note**
>
> L'alias del punto di accesso multiregionale e l'ARN non possono essere usati in modo intercambiabile.

Gli ARN dei punti di accesso multi-regione utilizzano il seguente formato:

`arn:aws:s3::account-id:accesspoint/MultiRegionAccessPoint_alias`

Di seguito sono riportati alcuni esempi di ARN dei punti di accesso multi-regione:

- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap` rappresenta il punto di accesso multi-regione con l'alias `mfzwi23gnjvgw.mrap`, di proprietà dell'Account AWS `123456789012`.
- `arn:aws:s3::123456789012:accesspoint/*` rappresenta tutti i punti di accesso multi-regione dell'account `123456789012`. Questo ARN corrisponde a tutti i punti di accesso multi-regione per l'account `123456789012`, ma non corrisponde ai punti di accesso Amazon S3 regionali perché l'ARN non include una Regione AWS. Per contro, l'ARN `arn:aws:s3:us-west-2:123456789012:accesspoint/*` corrisponde a tutti i punti di accesso Amazon S3 regionali della regione `us-west-2` per l'account `123456789012`, ma non corrisponde ad alcun punto di accesso multi-regione.

Gli ARN per gli oggetti a cui si accede tramite un punto di accesso multi-regione utilizzano il seguente formato:

`arn:aws:s3::account_id:accesspoint/MultiRegionAccessPoint_alias//key`

Come per gli ARN dei punti di accesso multi-regione, gli ARN degli oggetti a cui si accede tramite punti di accesso multi-regione non includono una Regione AWS. Ecco alcuni esempi.

- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//-01` rappresenta `-01`, accessibile tramite il punto di accesso multi-regione con l'alias `mfzwi23gnjvgw.mrap` di proprietà dell'account `123456789012`.

- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//*` rappresenta tutti gli oggetti a cui è possibile accedere tramite il punto di accesso multi-regione con l'alias `mfzwi23gnjvgw.mrap`, nell'account `123456789012`.
- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//-01/finance/*` rappresenta tutti gli oggetti a cui è possibile accedere in `-01/finance/` per il punto di accesso multi-regione con l'alias `mfzwi23gnjvgw.mrap` nell'account `123456789012`.

Nomi host del punto di accesso multi-regione

Puoi accedere ai dati in Amazon S3 tramite un punto di accesso multi-regione utilizzando il relativo nome host. Le richieste possono essere indirizzate a questo nome host dalla rete Internet pubblica. Se hai configurato uno o più gateway Internet per il punto di accesso multi-regione, è anche possibile indirizzare le richieste a questo nome host da un cloud privato virtuale (VPC). Per ulteriori informazioni sulla creazione di endpoint di interfaccia VPC da utilizzare con punti di accesso multi-regione, consultaConfigurazione di un punto di accesso multi-regione per l'utilizzo con AWS PrivateLink.

Per eseguire richieste tramite un punto di accesso multi-regione da un VPC mediante un endpoint VPC, puoi usare AWS PrivateLink. Quando esegui richieste su un punto di accesso multi-regione mediante AWS PrivateLink, non puoi utilizzare direttamente un sistema dei nomi di dominio (DNS) regionale specifico dell'endpoint che termina con *`region`*`.vpce.amazonaws.com`. Questo nome host non ha un certificato associato ad esso, quindi non può essere utilizzato direttamente. Puoi comunque utilizzare il sistema dei nomi di dominio (DNS) pubblico dell'endpoint VPC come destinazione di `CNAME` o `ALIAS`. In alternativa, puoi abilitare il sistema dei nomi di dominio (DNS) privato sull'endpoint e utilizzare il nome del sistema dei nomi di dominio (DNS) *`MultiRegionAccessPoint_alias`*`.accesspoint.s3-global.amazonaws.com` standard del punto di accesso multi-regione come descritto in questa sezione.

Quando effettui richieste all'API per le operazioni sui dati di Amazon S3, ad esempio `GetObject`, tramite un punto di accesso multi-regione, il nome host per la richiesta è come segue:

`MultiRegionAccessPoint_alias``.accesspoint.s3-global.amazonaws.com`

Ad esempio, per creare una richiesta `GetObject` tramite il punto di accesso multi-regione con l'alias `mfzwi23gnjvgw.mrap`, esegui una richiesta sul nome host `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`. La porzione `s3-global` del nome host indica che questo nome host non è per una regione specifica.

L'esecuzione di richieste tramite un punto di accesso multi-regione è simile all'esecuzione di richieste tramite un punto di accesso a una regione singola. È tuttavia importante essere consapevoli delle seguenti differenze:

- Gli ARN dei punti di accesso multi-regione non includono una Regione AWS. Seguono il formato `arn:aws:s3::account-id:accesspoint/MultiRegionAccessPoint_alias`.
- Per le richieste effettuate tramite operazioni API (tali richieste non richiedono l'uso di un ARN), i punti di accesso multi-regione utilizzano uno schema di endpoint diverso. Lo schema è `MultiRegionAccessPoint_alias.accesspoint.s3-global.amazonaws.com`, ad esempio `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`. Nota le differenze rispetto a un punto di accesso a una regione singola:
  - I nomi host del punto di accesso multi-regione utilizzano il proprio alias, non il nome del punto di accesso multi-regione.
  - I nomi host dei punti di accesso multi-regione non includono l'ID Account AWS del proprietario.
  - I nomi host dei punti di accesso multi-regione non includono una Regione AWS.
  - I nomi host del punto di accesso multi-regione includono `s3-global.amazonaws.com` invece di `s3.amazonaws.com`.
- Le richieste tramite punti di accesso multi-regione devono essere firmate utilizzando Signature Version 4A (Sigv4a). Quando utilizzi gli SDK AWS, l'SDK in uso converte automaticamente il formato SigV4 nel formato Sigv4A. Verifica pertanto che l'SDK AWS supporti il formato SigV4a come implementazione di firma utilizzata per firmare le richieste globali a livello di Regione AWS. Per ulteriori informazioni su SigV4A, consultare la sezione relativa alla firma di richieste API di AWS nella Riferimenti generali di AWS.

Punti di accesso multi-regione e Amazon S3 Transfer Acceleration

Amazon S3 Transfer Acceleration è una caratteristica che abilita trasferimenti di dati più rapidi a bucket. Transfer Acceleration è configurato a livello di singolo bucket. Per ulteriori informazioni su Transfer Acceleration, consulta Configurazione di trasferimenti veloci e sicuri di file con Amazon S3 Transfer Acceleration.

I punti di accesso multi-regione utilizzano un meccanismo di trasferimento accelerato simile a quello di Transfer Acceleration per l'invio di oggetti di grandi dimensioni tramite la rete AWS. Per questo motivo, non devi usare Transfer Acceleration quando invii richieste tramite un punto di accesso multi-regione. Questo miglioramento delle prestazioni di trasferimento viene incorporato automaticamente nel punto di accesso multi-regione.

Argomenti

Autorizzazioni

I punti di accesso multi-regione di Amazon S3 possono semplificare l'accesso ai dati per i bucket Amazon S3 in più Regioni AWS. I punti di accesso multi-regione sono endpoint globali denominati che possono essere utilizzati per eseguire operazioni su oggetti di accesso ai dati di Amazon S3, ad esempio `GetObject` e `PutObject`. Ogni punto di accesso multi-regione può disporre di autorizzazioni e controlli di rete distinti per qualsiasi richiesta eseguita tramite l'endpoint globale.

Ogni punto di accesso multi-regione può inoltre applicare una policy di accesso personalizzata che funziona in combinazione con la policy di bucket collegata al bucket sottostante. Affinché una richiesta abbia esito positivo, è necessario che tutti i seguenti elementi consentano l'operazione:

- Policy dei punti di accesso multi-regione
- Policy AWS Identity and Access Management (IAM) sottostante
- Policy di bucket sottostante (a cui viene indirizzata la richiesta)

È possibile configurare qualsiasi policy dei punti di accesso multi-regione per accettare richieste solo da gruppi o utenti IAM specifici. Per un esempio su come eseguire questa operazione, consulta l'esempio 2 in the section called “Esempi di policy dei punti di accesso multi-regione”. Per limitare l'accesso ai dati di Amazon S3 a una rete privata, puoi configurare la policy dei punti di accesso multi-regione in modo che accetti le richieste solo da un cloud privato virtuale (VPC).

Supponiamo, ad esempio, di creare una richiesta `GetObject` tramite un punto di accesso multi-regione utilizzando un utente denominato `AppDataReader` nel tuo account AWS. Per far sì che la richiesta non venga negata, l'utente `AppDataReader` deve ricevere l'autorizzazione `s3:GetObject`

dal punto di accesso multi-regione e da ogni relativo bucket sottostante. `AppDataReader` non sarà in grado di recuperare i dati dai bucket che non concedono questa autorizzazione.

> **Important**
>
> La delega del controllo dell'accesso di un bucket a una policy del punto di accesso multi-regione non modifica il comportamento del bucket a cui si accede direttamente tramite il nome del bucket o il nome della risorsa Amazon (ARN). Tutte le operazioni eseguite direttamente sul bucket continueranno a funzionare come prima. Le limitazioni incluse in una policy del punto di accesso multi-regione si applicano solo alle richieste effettuate tramite il corrispondente punto di accesso multi-regione.

Gestione dell'accesso pubblico a un punto di accesso multi-regione

I punti di accesso multi-regione supportano impostazioni di blocco dell'accesso pubblico indipendenti per ciascun punto di accesso. Quando crei un punto di accesso multi-regione puoi specificare le impostazioni di blocco dell'accesso pubblico applicabili.

> **Note**
>
> Tutte le impostazioni di blocco dell'accesso pubblico abilitate in Impostazioni di blocco dell'accesso pubblico per questo account (nel tuo account) o Impostazioni del Blocco dell'accesso pubblico per bucket esterni continuano a essere valide anche se le impostazioni indipendenti di blocco dell'accesso pubblico per il punto di accesso multi-regione sono disabilitate.

Per qualsiasi richiesta eseguita tramite un punto di accesso multi-regione, Amazon S3 valuta le impostazioni di Blocco dell'accesso pubblico Amazon S3 per:

- Punto di accesso multi-regione
- I bucket sottostanti (compresi i bucket esterni)
- L'account proprietario del punto di accesso multi-regione
- L'account proprietario dei bucket sottostanti (inclusi gli account esterni)

Se una di queste impostazioni indica che la richiesta deve essere bloccata, Amazon S3 rifiuta la richiesta. Per ulteriori informazioni sulla caratteristica di blocco dell'accesso pubblico di Amazon S3, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

> **Important**
>
> Per impostazione predefinita, tutte le impostazioni di Blocco dell'accesso pubblico Amazon S3 sono abilitate per i punti di accesso multi-regione. Devi disabilitare esplicitamente le impostazioni che non vuoi applicare a un punto di accesso multi-regione.
> Dopo la creazione del punto di accesso multi-regione, non puoi più modificare le relative impostazioni di blocco dell'accesso pubblico.

Visualizzazione delle impostazioni di Blocco dell'accesso pubblico Amazon S3 per un punto di accesso multi-regione

Per visualizzare le impostazioni di Blocco dell'accesso pubblico Amazon S3 per un punto di accesso multi-regione

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Multi-Region Access Points (Punti di accesso multi-regione).
3. Scegli il nome del punto di accesso multi-regione che desideri esaminare.
4. Scegliere la scheda Permissions (Autorizzazioni).
5. In Block Public Access settings for this Multi-Region Access Point (Impostazioni di Blocco dell'accesso pubblico per il punto di accesso multi-regione corrente), seleziona le impostazioni di Blocco dell'accesso pubblico Amazon S3 da applicare al tuo punto di accesso multi-regione.

> **Note**
>
> Dopo la creazione del punto di accesso multi-regione, non puoi modificare le impostazioni di Blocco dell'accesso pubblico Amazon S3. Pertanto, se intendi bloccare l'accesso pubblico, assicurati che le tue applicazioni funzionino correttamente senza accesso pubblico prima di creare un punto di accesso multi-regione.

Utilizzo di una policy dei punti di accesso multi-regione

Il seguente esempio di policy dei punti di accesso multi-regione consente a un utente IAM di visualizzare e scaricare file dal punto di accesso multi-regione. Per utilizzare questa policy di esempio, sostituisci *user input placeholders* con le tue informazioni.

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

Per associare la policy dei punti di accesso multi-regione al punto di accesso multi-regione specificato mediante AWS Command Line Interface (AWS CLI), utilizza il comando `put-multi-region-access-point-policy` seguente. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni. Ogni punto di accesso multi-regione può avere una sola policy, quindi una richiesta effettuata per l'operazione `put-multi-region-access-point-policy` sostituisce qualsiasi policy esistente associata al punto di accesso multi-regione specificato.

AWS CLI

```
aws s3control put-multi-region-access-point-policy
--account-id 111122223333
--details { "Name": "DOC-EXAMPLE-BUCKET-MultiRegionAccessPoint",
 "Policy": "{ \"Version\": \"2012-10-17\", \"Statement\": { \"Effect\":
 \"Allow\", \"Principal\": { \"AWS\": \"arn:aws:iam::111122223333:root
\" }, \"Action\": [\"s3:ListBucket\", \"s3:GetObject\"], \"Resource\":
```

```
 [ \"arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias",
 \"arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias/object/*
\" ] } }" }
```

Per esaminare i risultati dell'operazione precedente, utilizza il comando seguente:

AWS CLI

```
aws s3control describe-multi-region-access-point-operation
--account-id 111122223333
--request-token-arn requestArn
```

Per recuperare la policy dei punti di accesso multi-regione, utilizza il comando seguente:

AWS CLI

```
aws s3control get-multi-region-access-point-policy
--account-id 111122223333
--name=DOC-EXAMPLE-BUCKET-MultiRegionAccessPoint
```

Modifica della policy dei punti di accesso multi-regione

La policy dei punti di accesso multi-regione (scritta in JSON) fornisce l'accesso all'archiviazione ai bucket Amazon S3 utilizzati con questo punto di accesso multi-regione. Puoi consentire o negare a principali specifici di eseguire varie azioni sul tuo punto di accesso multi-regione. Quando una richiesta viene instradata a un bucket tramite il punto di accesso multi-regione, si applicano le policy di accesso per il punto di accesso multi-regione e per il bucket. La policy di accesso più restrittiva ha sempre la precedenza.

> **Note**
>
> Se un bucket contiene oggetti di proprietà di altri account, la policy dei punti di accesso multi-regione non si applica agli oggetti di proprietà di altri Account AWS.

Dopo aver applicato una policy dei punti di accesso multi-regione, tale policy non può essere eliminata. È possibile modificare la policy o creare una nuova policy che sovrascriva quella esistente.

Per esaminare la policy dei punti di accesso multi-regione

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Multi-Region Access Points (Punti di accesso multi-regione).
3. Seleziona il nome del punto di accesso multi-regione per il quale desideri modificare la policy.
4. Scegliere la scheda Permissions (Autorizzazioni).
5. Scorri verso il basso fino alla sezione Multi-Region Access Point policy (Policy del punto di accesso multi-regione). Scegli Edit (Modifica) per aggiornare la policy (in JSON).
6. Viene visualizzata la pagina Edit Multi-Region Access Point policy (Modifica policy del punto di accesso multi-regione). Puoi immettere la policy direttamente nel campo di testo oppure puoi scegliere Add statement (Aggiungi istruzione) per selezionare gli elementi della policy da un elenco a discesa.

> **Note**
>
> La console visualizza automaticamente il nome della risorsa Amazon (ARN) del punto di accesso multi-regione che può essere utilizzato nella policy. Per degli esempi di policy dei punti di accesso multi-regione, consulta the section called “Esempi di policy dei punti di accesso multi-regione”.

Esempi di policy dei punti di accesso multi-regione

I punti di accesso multi-regione di Amazon S3 supportano le policy di risorseAWS Identity and Access Management (IAM). È possibile utilizzare queste policy per controllare l'utilizzo del punto di accesso multi-regione per risorsa, utente o altre condizioni. Affinché un'applicazione o un utente possa accedere agli oggetti tramite un punto di accesso multi-regione, sia il punto di accesso multi-regione che il bucket sottostante devono consentire la stessa richiesta.

Per consentire lo stesso accesso sia al punto di accesso multi-regione che al bucket sottostante, esegui una delle seguenti operazioni:

- (Consigliato) Per semplificare i controlli di accesso quando si utilizza un punto di accesso multi-regione di Amazon S3, delega il controllo dell'accesso per il bucket Amazon S3 al punto di accesso

multi-regione. Per un esempio su come eseguire questa operazione, consulta l'esempio 1 in questa sezione.

- Aggiungi le stesse autorizzazioni contenute nella policy del punto di accesso multi-regione alla policy di bucket sottostante.

> **Important**
>
> La delega del controllo dell'accesso di un bucket a una policy del punto di accesso multi-regione non modifica il comportamento del bucket a cui si accede direttamente tramite il nome del bucket o il nome della risorsa Amazon (ARN). Tutte le operazioni eseguite direttamente sul bucket continueranno a funzionare come prima. Le limitazioni incluse in una policy del punto di accesso multi-regione si applicano solo alle richieste effettuate tramite il corrispondente punto di accesso multi-regione.

Example 1: delega dell'accesso a specifici punti di accesso multi-regione nella policy di bucket (per lo stesso account o per più account)

La seguente policy di esempio per i bucket consente l'accesso completo a livello di bucket a un punto di accesso multi-regione specifico. Questo significa che tutto l'accesso a questo bucket è controllato dalle policy associate al punto di accesso multi-regione. Si consiglia di configurare i bucket in questo modo per tutti i casi d'uso che non richiedono l'accesso diretto al bucket. È possibile utilizzare questa struttura di policy di bucket per i punti di accesso multi-regione nello stesso account o in un altro account.

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS": "*" },
        "Action" : "*",
        "Resource" : [ "Bucket ARN", "Bucket ARN/*"],
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointArn" : "MultiRegionAccessPoint_ARN" }
        }
    }]
}
```

> **Note**
>
> Se sono presenti più punti di accesso multi-regione a cui concedi l'accesso, assicurati di specificare ogni punto di accesso multi-regione.

Example 2: concessione a un account dell'accesso a un punto di accesso multi-regione nella policy del punto di accesso multi-regione

La seguente policy del punto di accesso multi-regione consente all'account *123456789012* di elencare e leggere gli oggetti contenuti nel punto di accesso multi-regione definito con *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: policy del punto di accesso multi-regione che consente l'elenco dei bucket

La seguente policy del punto di accesso multi-regione consente all'account *123456789012* di elencare gli oggetti contenuti nel punto di accesso multi-regione definito con *MultiRegionAccessPoint_ARN*.

```
{
   "Version":"2012-10-17",
```

```
   "Statement": [
      {
       "Effect": "Allow",
       "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/JohnDoe"
        },
        "Action": "s3:ListBucket",
        "Resource": "MultiRegionAccessPoint_ARN"
       }
    ]
}
```

Restrizioni e limitazioni dei punti di accesso multi-regione

I punti di accesso multi-regione di Amazon S3 presentano le seguenti restrizioni e limitazioni:

- Nomi dei punti di accesso multi-regione:
  - Deve essere unico all'interno di un singolo AWS account
  - Devono iniziare con un numero o una lettera minuscola
  - Devono contenere da 3 a 50 caratteri
  - Non possono iniziare o terminare con un trattino (`-`).
  - Non possono contenere caratteri di sottolineatura (`_`), lettere maiuscole o punti (`.`).
  - Non possono essere modificati dopo la creazione.
- Gli alias dei punti di accesso multi-regione vengono generati da Amazon S3 e non possono essere modificati o riutilizzati.
- Non puoi accedere ai dati tramite un punto di accesso multi-regione utilizzando endpoint gateway. Puoi invece accedere ai dati tramite un punto di accesso multi-regione utilizzando endpoint di interfaccia. Per utilizzarlo AWS PrivateLink, devi creare endpoint VPC. Per ulteriori informazioni, consulta Configurazione di un punto di accesso multi-regione per l'utilizzo con AWS PrivateLink.
- Per utilizzare punti di accesso multiregionali con Amazon CloudFront, devi configurare il punto di accesso multiregionale come tipo di `Custom Origin` distribuzione. Per ulteriori informazioni sui vari tipi di origine, consulta Usare origini diverse con CloudFront le distribuzioni. Per ulteriori informazioni sull'utilizzo di punti di accesso multiregionali con Amazon CloudFront, consulta Creazione di un'applicazione attiva-attiva e basata sulla prossimità in più regioni sul blog Storage.AWS
- Requisiti minimi per i punti di accesso multi-regione:
  - Transport Layer Security (TLS) v1.2

- Signature Version 4 (SigV4A)

  I punti di accesso multi-regione supportano Signature Version 4A. Questa versione di SigV4 consente di firmare le richieste per più Regioni AWS. Questa caratteristica è utile nelle operazioni API che potrebbero comportare l'accesso ai dati da una tra più regioni. Quando utilizzi un AWS SDK, fornisci le tue credenziali e le richieste ai punti di accesso multiregionali utilizzeranno la versione 4A di Signature senza configurazioni aggiuntive. Assicurati di verificare la compatibilità del tuo SDK AWS con l'algoritmo SigV4a. Per ulteriori informazioni su SigV4a, consulta Firmare le richieste API in. AWSRiferimenti generali di AWS

  > **Note**
  >
  > Per utilizzare Sigv4A con credenziali di sicurezza temporanee, ad esempio quando si utilizzano ruoli (IAM), è possibile richiedere le credenziali temporanee da un endpoint Regional AWS Identity and Access Management (). AWS Security Token Service AWS STS Se richiedi credenziali temporanee all' AWS STS endpoint globale (`sts.amazonaws.com`), devi prima impostare la compatibilità regionale dei token di sessione affinché l'endpoint globale sia valido in tutti. Regioni AWSPer ulteriori informazioni, consulta Managing AWS STS in an Regione AWS nella IAM User Guide.

- I punti di accesso multi-regione non supportano richieste anonime.
- Limitazioni dei punti di accesso multi-regione:
  - Il protocollo IPv6 non è supportato.
  - I bucket Amazon S3 su Outposts non sono supportati.
  - I punti di accesso multiregione supportano le operazioni di copia utilizzando punti di accesso multiregione solo come destinazione quando si utilizza l'ARN del punto di accesso multiregionale.
  - La funzionalità Operazioni in batch S3 non è supportata.
- Alcuni AWS SDK non sono supportati. Per confermare quali AWS SDK sono supportati per i punti di accesso multiregionali, consulta Compatibilità con gli SDK. AWS
- Le Service Quotas per i punti di accesso multi-regione sono indicate di seguito:
  - È previsto un massimo di 100 punti di accesso multi-regione per account.
  - Esiste un limite di 17 regioni per un singolo punto di accesso multi-regione.
- Dopo aver creato un punto di accesso multi-regione, non puoi aggiungere, modificare o rimuovere i bucket dalla relativa configurazione. Per modificare i bucket, devi eliminare l'intero punto di

accesso multi-regione e crearne uno nuovo. Se viene eliminato un bucket multi-account nel punto di accesso multi-regione, l'unico modo per ricollegarlo è ricreare il bucket utilizzando lo stesso nome e la stessa regione in tale account.

- I bucket sottostanti (nello stesso account) utilizzati in un punto di accesso multi-regione possono essere eliminati solo dopo aver eliminato il punto di accesso multi-regione associato.
- Tutte le richieste del piano di controllo (control-plane) per creare o mantenere punti di accesso multi-regione devono essere instradate alla regione `US West (Oregon)`. Per richieste sul piano dati del punto di accesso multi-regione, non è necessario specificare le regioni.
- Per il piano di controllo (control-plane) di failover del punto di accesso multi-regione, le richieste devono essere instradate a una delle cinque regioni supportate seguenti:
  - `US East (N. Virginia)`
  - `US West (Oregon)`
  - `Asia Pacific (Sydney)`
  - `Asia Pacific (Tokyo)`
  - `Europe (Ireland)`
- Il tuo punto di accesso multiregionale supporta solo i bucket seguenti: Regioni 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)`

Instradamento della richiesta tramite punto di accesso multi-regione

Quando effettui una richiesta tramite un punto di accesso multi-regione, Amazon S3 individua i bucket associati al punto di accesso multi-regione più vicini. Amazon S3 indirizza quindi la richiesta a quel bucket, indipendentemente dalla regione AWS in cui si trova.

Dopo che il punto di accesso multi-regione instrada la richiesta al bucket più vicino, Amazon S3 elabora la richiesta come se fosse stata eseguita direttamente su tale bucket. I punti di accesso multi-regione non rilevano il contenuto dei dati di un bucket Amazon S3. Pertanto, il bucket che riceve la richiesta potrebbe non contenere i dati richiesti. Per creare set di dati coerenti nei bucket Amazon S3 associati a un punto di accesso multi-regione, puoi configurare la replica tra regioni di Amazon S3 (CRR). Quindi qualsiasi bucket può soddisfare la richiesta correttamente.

Amazon S3 indirizza le richieste dei punti di accesso multi-regione in base alle seguenti regole:

- Amazon S3 ottimizza le richieste da evadere in base alla prossimità. Esamina i bucket supportati dal punto di accesso multi-regione e inoltra la richiesta al bucket più vicino.
- Se la richiesta specifica una risorsa esistente (ad esempio `GetObject`), Amazon S3 non considera il nome dell'oggetto durante l'adempimento della richiesta. Ciò significa che un oggetto potrebbe esistere in un bucket nel punto di accesso multi-regione, ma la richiesta verrà instradata a un bucket che non contiene l'oggetto. Questo scenario restituirà un messaggio di errore 404 al client.

  Per evitare errori 404, ti consigliamo di configurare la replica tra regioni di Amazon S3 (S3 CRR) per i bucket. La replica consente infatti di risolvere il problema potenziale che nasce quando l'oggetto desiderato si trova in un bucket del punto di accesso multi-regione, ma non si trova nel bucket specifico a cui è stata instradata la richiesta. Per maggiori informazioni sulla configurazione della replica, consulta Configurazione della replica per l'utilizzo con punti di accesso multi-regione.

  Per garantire che le richieste vengano soddisfatte utilizzando gli oggetti specifici desiderati, è inoltre consigliabile attivare il controllo delle versioni dei bucket e includere gli ID delle versioni nelle richieste. In questo modo sei sicuro di disporre della versione corretta dell'oggetto che stai cercando. I bucket con la funzione di controllo delle versioni abilitata consentono di ripristinare gli oggetti che sono stati sovrascritti per errore. Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

- Se la richiesta prevede di creare una risorsa (ad esempio `PutObject` o `CreateMultipartUpload`), Amazon S3 la esegue utilizzando il bucket più vicino. Ad esempio, considera un'azienda di video che vuole supportare i caricamenti video da qualsiasi parte del mondo. Quando un utente invia una richiesta PUT al punto di accesso multi-regione, l'oggetto viene inserito nel bucket più vicino. Per rendere il video caricato disponibile ad altre persone in tutto il mondo per il download con la latenza più bassa, puoi utilizzare la funzionalità di replica tra regioni di S3 (S3 CRR) con replica bidirezionale. L'uso di questa funzionalità con la replica tra regioni mantiene sincronizzato il contenuto di tutti i bucket associati al punto di accesso multi-regione. Per ulteriori informazioni sulla replica con i punti di accesso multi-regione, consulta Configurazione della replica per l'utilizzo con punti di accesso multi-regione.

Controlli di failover dei punti di accesso multi-regione Amazon S3

Con i controlli di failover dei punti di accesso multi-regione di Amazon S3, puoi mantenere la continuità aziendale durante le interruzioni del traffico regionale, dotando al contempo le tue applicazioni di un'architettura multi-regione per soddisfare le esigenze di conformità e ridondanza. Se il traffico regionale subisce interruzioni, puoi utilizzare i controlli di failover dei punti di accesso multi-regione per selezionare le Regioni AWS di riferimento per il punto di accesso multi-regione Amazon S3 che elaboreranno le richieste di accesso ai dati e di archiviazione.

Per supportare il failover, è possibile configurare il punto di accesso multi-regione in una configurazione attiva-passiva, con il traffico che fluisce verso la regione attiva in condizioni normali e una regione passiva in standby per il failover.

Ad esempio, per eseguire il failover su una Regione AWS di tua scelta, sposta il traffico dalla tua regione principale (attiva) alla tua regione secondaria (passiva). In una configurazione attiva-passiva come questa, un bucket è attivo e accetta traffico, mentre l'altro bucket è passivo e non accetta traffico. Il bucket passivo viene utilizzato per il ripristino di emergenza. Quando si avvia il failover, tutto il traffico (ad esempio le richieste `GET` o `PUT`) viene indirizzato al bucket nello stato attivo (in una regione) e allontanato dal bucket nello stato passivo (in un'altra regione).

Se la replica tra regioni di S3 (S3 CRR) è abilitata con regole di replica bidirezionale, puoi mantenere sincronizzati i bucket durante un failover. Inoltre, se hai abilitato la replica CRR in una configurazione attiva-attiva, i punti di accesso multi-regione Amazon S3 possono anche recuperare i dati dalla posizione del bucket più vicina, migliorando le prestazioni delle applicazioni.

Supporto di Regione AWS

Con i controlli di failover dei punti di accesso multi-regione Amazon S3, i tuoi bucket S3 possono trovarsi in una qualsiasi delle 17 regioni in cui sono supportati i punti di accesso multi-regione. È possibile avviare il failover in due regioni qualsiasi contemporaneamente.

> **Note**
>
> Sebbene il failover venga avviato solo tra due regioni contemporaneamente, è possibile aggiornare separatamente gli stati di instradamento per più regioni contemporaneamente nel punto di accesso multi-regione.

I seguenti argomenti illustrano come utilizzare e gestire i controlli di failover dei punti di controllo multi-regione Amazon S3.

Argomenti

- Stati di instradamenti dei punti di accesso multi-regione Amazon S3
- Utilizzo dei controlli di failover dei punti di accesso multi-regione Amazon S3
- Errori dei controlli di failover dei punti di accesso multi-regione Amazon S3

Stati di instradamenti dei punti di accesso multi-regione Amazon S3

La configurazione del failover dei punti di accesso multi-regione di Amazon S3 determina lo stato di instradamento delle Regioni AWS utilizzate con il punto di accesso multi-regione. Puoi configurare il tuo punto di accesso multi-regione Amazon S3 in modo che sia in uno stato attivo-attivo o attivo-passivo.

- Attivo-attivo: in una configurazione attiva-attiva, tutte le richieste vengono inviate automaticamente alla Regione AWS del punto di accesso multi-regione più vicino. Dopo che il punto di accesso multi-regione è stato configurato con stato attivo, tutte le regioni possono ricevere traffico. Se si verifica un'interruzione del traffico in una configurazione attiva-attiva, il traffico di rete verrà automaticamente reindirizzato a una delle regioni attive.
- Attivo-passivo: in una configurazione attiva-passiva, le regioni attive nel punto di accesso multi-regione ricevono traffico e quelle passive no. Se intendi utilizzare i controlli di failover S3 per avviare il failover in una situazione di emergenza, configura i tuoi punti di accesso multi-regione

in una configurazione attiva-passiva mentre esegui i test e la pianificazione del ripristino di emergenza.

Utilizzo dei controlli di failover dei punti di accesso multi-regione Amazon S3

Questa sezione illustra come gestire e utilizzare i controlli di failover dei punti di accesso Amazon S3 utilizzando la AWS Management Console.

Esistono due controlli di failover nella sezione Failover configuration (Configurazione failover) nella pagina dei dettagli del punto di accesso multi-regione nella AWS Management Console: Edit routing status (Modifica stato di instradamento) e Failover. Puoi utilizzare questi controlli nel modo seguente:

- Edit routing status (Modifica stato di instradamento): puoi modificare manualmente gli stati di instradamento di un massimo di 17 Regioni AWS in una singola richiesta per il tuo punto di accesso multi-regione scegliendo Edit routing status (Modifica stato di instradamento). È possibile utilizzare Edit routing status (Modifica stato di instradamento) per i seguenti scopi:
  - Per impostare o modificare gli stati di instradamento di una o più regioni nel punto di accesso multi-regione
  - Per creare una configurazione di failover per il punto di accesso multi-regione configurando due regioni in modo che siano in uno stato attivo-passivo
  - Per eseguire manualmente il failover delle regioni
  - Per scambiare manualmente il traffico tra regioni
- Failover: quando si avvia il failover scegliendo Failover, si aggiornano solo gli stati di instradamento di due regioni già configurate per essere in uno stato attivo-passivo. Durante un failover avviato scegliendo Failover, gli stati di instradamento tra le due regioni vengono scambiati automaticamente.

Modifica dello stato di instradamento delle regioni nel punto di accesso multi-regione

È possibile aggiornare manualmente gli stati di instradamento di un massimo di 17Regioni AWS in un'unica richiesta per il punto di accesso multi-regione scegliendo Edit routing status (Modifica stato di instradamento) nella sezione Failover configuration (Configurazione failover) nella pagina dei dettagli del punto di accesso multi-regione. Tuttavia, quando si avvia il failover scegliendo Failover, si aggiornano solo gli stati di instradamento di due regioni già configurate per essere in uno stato attivo-passivo. Durante un failover avviato scegliendo Failover, gli stati di instradamento tra le due regioni vengono scambiati automaticamente.

È possibile utilizzare Edit routing status (Modifica stato di instradamento) (come descritto nella procedura seguente) per i seguenti scopi:

- Per impostare o modificare gli stati di instradamento di una o più regioni nel punto di accesso multi-regione
- Per creare una configurazione di failover per il punto di accesso multi-regione configurando due regioni in modo che siano in uno stato attivo-passivo
- Per eseguire manualmente il failover delle regioni
- Per scambiare manualmente il traffico tra regioni

Utilizzo della console S3

Per aggiornare lo stato di instradamento delle regioni nel punto di accesso multi-regione

1. Accedere alla Console di gestione AWS.
2. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
3. Nel pannello di navigazione a sinistra, scegli Multi-Region Access Points (Punti di accesso multi-regione).
4. Scegli il punto di accesso multi-regione da aggiornare.
5. Scegli la scheda Replication and failover (Replica e failover).
6. Seleziona una o più regioni di cui desideri modificare lo stato di instradamento.

   > **Note**
   >
   > Per avviare il failover, almeno una Regione AWS deve essere designata come attiva e una regione deve essere designata come passiva nel punto di accesso multi-regione.

7. Scegli Edit routing status (Modifica stato di instradamento).
8. Nella finestra di dialogo visualizzata, seleziona Active (Attivo) o Passive (Passivo) per l'opzione Routing status (Stato instradamento) per ciascuna regione.

   Uno stato attivo consente di indirizzare il traffico verso la regione. Uno stato passivo impedisce che qualsiasi traffico venga indirizzato verso la regione.

Se si sta creando una configurazione di failover per il punto di accesso multi-regione o si avvia il failover, almeno una Regione AWS deve essere designata come attiva e una regione deve essere designata come passiva nel punto di accesso multi-regione.

9. Scegli Save routing status (Salva stato di instradamento). Il reindirizzamento del traffico richiede circa 2 minuti.

Dopo aver inviato lo stato di instradamento della Regioni AWS del punto di accesso multi-regione, è possibile verificare le modifiche di tale stato. Per verificare queste modifiche, vai su Amazon CloudWatch all'indirizzo https://console.aws.amazon.com/cloudwatch/ per monitorare lo spostamento del traffico di richieste di dati Amazon S3 (ad esempio richieste `GET` e `PUT`) tra regioni attive e passive. Le connessioni esistenti non verranno interrotte durante il failover. Le connessioni esistenti continueranno fino a raggiungere lo stato di operazione riuscita o errore.

Utilizzo di AWS CLI

> **Note**
>
> È possibile eseguire comandi AWS CLI di instradamento per i punti di accesso multi-regione su una qualsiasi di queste cinque regioni:
>
> - `ap-southeast-2`
> - `ap-northeast-1`
> - `us-east-1`
> - `us-west-2`
> - `eu-west-1`

Il seguente comando di esempio aggiorna la configurazione di instradamento corrente per i punti di accesso multi-regione. Per aggiornare lo stato attivo o passivo di un bucket, imposta il valore `TrafficDialPercentage` su `100` per attivo e su `0` per passivo. In questo esempio, *`DOC-EXAMPLE-BUCKET-1`* è impostato su attivo e *`DOC-EXAMPLE-BUCKET-2`* su passivo. Per utilizzare questo comando di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

```
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
```

Il seguente comando di esempio recupera la configurazione di instradamento aggiornata per i punti di accesso multi-regione. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 111122223333
--mrap MultiRegionAccessPoint_ARN
```

Avvio del failover

Quando avvii il failover scegliendo Failover nella sezione Failover configuration (Configurazione failover) nella pagina dei dettagli del punto di accesso multi-regione, il traffico delle richieste Amazon S3 viene automaticamente spostato su una Regione AWS alternativa. Il processo di failover viene completato entro 2 minuti.

È possibile avviare un failover su due Regioni AWS contemporaneamente (delle 17 regioni in cui sono supportati i punti di accesso multi-regione). Gli eventi di failover vengono quindi registrati in AWS CloudTrail. Al termine del failover, puoi monitorare il traffico di Amazon S3 e qualsiasi aggiornamento dell'instradamento del traffico nella nuova regione attiva in Amazon CloudWatch.

> **Important**
>
> Per mantenere sincronizzati tutti i metadati e gli oggetti tra i bucket durante la replica dei dati, si consiglia di creare regole di replica bidirezionali e abilitare la sincronizzazione delle modifiche della replica prima di configurare i controlli di failover.
> Le regole di replica bidirezionale aiutano a garantire che quando i dati vengono scritti nel bucket Amazon S3, il traffico viene poi replicato nuovamente nel bucket di origine. La sincronizzazione delle modifiche delle repliche aiuta a garantire che i metadati degli oggetti siano sincronizzati anche tra i bucket durante la replica bidirezionale.
> Per maggiori informazioni sulla configurazione della replica per supportare il failover, consulta the section called “Replica del bucket”.

Per avviare il failover tra bucket replicati

1. Accedere alla Console di gestione AWS.
2. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
3. Nel pannello di navigazione a sinistra, scegli Multi-Region Access Points (Punti di accesso multi-regione).
4. Scegli il punto di accesso multi-regione da utilizzare per avviare il failover.
5. Scegli la scheda Replication and failover (Replica e failover).
6. Scorri verso il basso fino alla sezione Failover configuration (Configurazione failover) e selezionane due Regioni AWS.

   > **Note**
   >
   > Per avviare il failover, almeno una Regione AWS deve essere designata come attiva e una regione deve essere designata come passiva nel punto di accesso multi-regione. Uno stato attivo consente di indirizzare il traffico verso una regione. Uno stato passivo impedisce che qualsiasi traffico venga indirizzato verso la regione.

7. Scegli Failover.
8. Nella finestra di dialogo, scegli di nuovo Failover per avviare il processo di failover.
   Durante questo processo, gli stati di instradamento delle due regioni vengono scambiati automaticamente. Tutto il nuovo traffico viene indirizzato alla regione che diventa attiva e il traffico smette di essere indirizzato verso la regione che diventa passiva. Il reindirizzamento del traffico richiede circa 2 minuti.

   Dopo aver avviato il processo di failover, puoi verificare le variazioni a livello di traffico.
   Per verificare queste modifiche, vai su Amazon CloudWatch all'indirizzo https://console.aws.amazon.com/cloudwatch/ per monitorare lo spostamento del traffico di richieste di dati Amazon S3 (ad esempio richieste `GET` e `PUT`) tra regioni attive e passive. Le connessioni esistenti non verranno interrotte durante il failover. Le connessioni esistenti continueranno fino a raggiungere lo stato di operazione riuscita o errore.

Visualizzazione dei controlli di instradamento del punto di accesso multi-regione Amazon S3

Utilizzo della console S3

Per visualizzare i controlli di instradamento per il punto di accesso multi-regione Amazon S3

1. Accedere alla Console di gestione AWS.
2. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
3. Nel pannello di navigazione a sinistra, scegli Multi-Region Access Points (Punti di accesso multi-regione).
4. Scegli il punto di accesso multi-regione che desideri esaminare.
5. Scegli la scheda Replication and failover (Replica e failover). Questa pagina mostra i dettagli e il riepilogo della configurazione dell'instradamento per il punto di accesso multi-regione, le regole di replica associate e i parametri di replica. Puoi vedere lo stato del instradamento delle tue regioni nella sezione Failover configuration (Configurazione failover).

Utilizzo di AWS CLI

Il seguente comando AWS CLI di esempio recupera la configurazione corrente dell'instradamento del punto di accesso multi-regione per la regione specificata. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 111122223333
--mrap MultiRegionAccessPoint_ARN
```

> **Note**
>
> Questo comando può essere eseguito solo su queste cinque regioni:
>
> - ap-southeast-2
> - ap-northeast-1
> - us-east-1
> - us-west-2
> - eu-west-1

Errori dei controlli di failover dei punti di accesso multi-regione Amazon S3

Quando si aggiorna la configurazione di failover per il punto di accesso multi-regione, è possibile che si verifichi uno dei seguenti errori:

- HTTP 400 - Richiesta non valida): questo errore si può verificare se si inserisce un ARN non valido per un punto di accesso multi-regione durante l'aggiornamento della configurazione del failover. Puoi confermare l'ARN del punto di accesso multi-regione facendo riferimento alla policy del punto di accesso multi-regione in questione. Per esaminare o aggiornare la policy del punto di accesso multi-regione, consulta Modifica della policy dei punti di accesso multi-regione. Questo errore può verificarsi anche se si utilizza una stringa vuota o una stringa casuale durante l'aggiornamento dei controlli di failover del punto di accesso multi-regione Amazon S3. Assicurati di usare il seguente formato di ARN per punti di accesso multi-regione:

  `arn:aws:s3::account-id:accesspoint/MultiRegionAccessPoint_alias`

- HTTP 503 Slow Down (HTTP 503 - Rallentamento): questo errore si verifica se si inviano troppe richieste in un breve periodo di tempo. Le richieste rifiutate genereranno un errore.
- HTTP 409 Conflict (HTTP 409 - Conflitto): questo errore si verifica quando due o più richieste di aggiornamento simultanee della configurazione dell'instradamento sono destinate a un unico punto di accesso multi-regione. La prima richiesta ha esito positivo, ma tutte le altre richieste hanno esito negativo e ciò genera un errore.
- HTTP 405 Method Not Allowed (HTTP 405 - metodo non concesso): questo errore si verifica quando si seleziona un punto di accesso multi-regione solo con una Regione AWS all'avvio del failover. È necessario selezionare due regioni prima di poter avviare il failover. In caso contrario, viene restituito un errore.

Configurazione della replica per l'utilizzo con punti di accesso multi-regione

Quando effettui una richiesta all'endpoint di un punto di accesso multi-regione, Amazon S3 instrada automaticamente la richiesta al bucket più vicino. Per questa decisione, Amazon S3 non prende in considerazione il contenuto della richiesta. Se esegui una richiesta `GET` per un oggetto, la richiesta potrebbe essere instradata a un bucket che non dispone di una copia dell'oggetto. In questo caso, riceverai un errore con il codice di stato HTTP 404 (Non trovato). Per ulteriori informazioni sull'instradamento delle richieste ai punti di accesso multi-regione, consulta the section called “Instradamento della richiesta”.

Se desideri che il punto di accesso multi-regione sia in grado di recuperare l'oggetto indipendentemente dal bucket che riceve la richiesta, devi configurare la replica tra regioni di Amazon S3 (CRR).

Ad esempio, considera un punto di accesso multi-regione con tre bucket:

- Un bucket denominato `my-bucket-usw2` nella regione `us-west-2` che contiene l'oggetto `my-image.jpg`
- Un bucket denominato `my-bucket-aps1` nella regione `ap-south-1` che contiene l'oggetto `my-image.jpg`
- Un bucket denominato `my-bucket-euc1` nella regione `eu-central-1` che non contiene l'oggetto `my-image.jpg`

In questa situazione, se esegui una richiesta `GetObject` per l'oggetto`my-image.jpg`, il successo della richiesta dipende dal bucket che la riceve. Poiché Amazon S3 non considera il contenuto della richiesta, potrebbe instradare la richiesta `GetObject` al bucket `my-bucket-euc1` se è il bucket più vicino che risponde. Anche se l'oggetto si trova in un bucket nel punto di accesso multi-regione, otterrai un errore 404 (non trovato) perché il singolo bucket che ha ricevuto la richiesta non ha l'oggetto.

L'attivazione della replica tra regioni (CRR) consente di evitare questo risultato. Con le regole di replica appropriate, l'oggetto `my-image.jpg` viene copiato nel bucket `my-bucket-euc1`. Pertanto, se Amazon S3 instrada la tua richiesta a quel bucket, ora puoi recuperare l'oggetto.

La replica funziona normalmente con i bucket assegnati a un punto di accesso multi-regione. Amazon S3 non esegue alcuna gestione speciale con i bucket che si trovano in punti di accesso multi-regione. Per ulteriori informazioni sulla configurazione della replica nei bucket, consulta Configurazione della replica in tempo reale.

Suggerimenti per l'utilizzo della replica con i punti di accesso multi-regione

Per ottimizzare le prestazioni di replica in caso di utilizzo dei punti di accesso multi-regione, consigliamo quanto segue:

- Configurare la funzionalità di controllo del tempo di replica di S3 (S3 RTC). Per eseguire la replica dei dati in regioni diverse in un arco di tempo prevedibile, puoi utilizzare S3 RTC. S3 RTC replica il 99,99% dei nuovi oggetti archiviati in Amazon S3 entro 15 minuti, secondo un Accordo sul Livello di Servizio (SLA). Per ulteriori informazioni, consulta the section called “Utilizzo di S3 Replication

Time Control". Non vi sono costi aggiuntivi per S3 RTC. Per ulteriori informazioni, consulta la pagina Prezzi di Amazon S3.

- Utilizza la replica bidirezionale per supportare la sincronizzazione dei bucket quando questi vengono aggiornati tramite il punto di accesso multi-regione. Per ulteriori informazioni, consulta the section called "Creare regole di replica bidirezionale per il punto di accesso multi-regione".
- Crea punti di accesso multi-regione multi-account per replicare i dati in bucket in Account AWS distinti. Questo approccio prevede la separazione a livello di account, in modo che i dati possano essere accessibili e replicati su diversi account in regioni diverse dal bucket di origine. La configurazione di punti di accesso multi-regione multi-account non comporta costi aggiuntivi. Se sei proprietario di un bucket ma non possiedi il punto di accesso multi-regione, paghi solo i costi di richiesta e trasferimento dei dati. I proprietari di punti di accesso multi-regione pagano i costi di instradamento dei dati e accelerazione di Internet. Per ulteriori informazioni, consulta i Prezzi di Amazon S3.
- Abilita la sincronizzazione delle modifiche delle repliche per ogni regola di replica per mantenere sincronizzate anche le modifiche dei metadati degli oggetti. Per ulteriori informazioni, consulta Abilitazione della sincronizzazione delle modifiche alla replica.
- Abilita i parametri di Amazon CloudWatch per monitorare gli eventi di replica. Si applicano le tariffe valide per i parametri CloudWatch. Per ulteriori informazioni, consulta Prezzi di Amazon CloudWatch.

Argomenti

Creare una regola di replica unidirezionale per il punto di accesso multi-regione

Le regole di replica consentono la copia asincrona e automatica di oggetti tra bucket. Una regola di replica unidirezionale consente di garantire che i dati vengano replicati completamente da un bucket di origine in una Regione AWS a un bucket di destinazione in un'altra regione. Quando è configurata la replica unidirezionale, viene creata una regola di replica dal bucket di origine (DOC-EXAMPLE-BUCKET-1) al bucket di destinazione (DOC-EXAMPLE-BUCKET-2). Come tutte le regole di replica, puoi applicare la regola di replica unidirezionale all'intero bucket Amazon S3 o a un sottoinsieme di oggetti filtrati per prefisso o tag oggetto.

> **Important**
>
> Ti consigliamo di utilizzare la replica unidirezionale se gli utenti utilizzeranno solo gli oggetti nei bucket di destinazione. Se gli utenti caricheranno o modificheranno gli oggetti nei bucket di destinazione, utilizza la replica bidirezionale per mantenere sincronizzati tutti i bucket. Ti consigliamo anche di utilizzare la replica bidirezionale se prevedi di usare il punto di accesso multi-regione per il failover. Per configurare la replica bidirezionale, consulta the section called “Creare regole di replica bidirezionale per il punto di accesso multi-regione”.

Per creare una regola di replica bidirezionale per il punto di accesso multi-regione

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Multi-Region Access Points (Punti di accesso multi-regione).
3. Scegli il nome del punto di accesso multi-regione.
4. Scegli la scheda Replication and failover (Replica e failover).
5. Scorri verso il basso fino alla sezione Replication rules (Regole di replica) e quindi scegli Create replication rules (Crea regole di replica). Assicurati di disporre di autorizzazioni sufficienti per creare la regola di replica; in caso contrario, il controllo delle versioni verrà disabilitato.

   > **Note**
   >
   > Puoi creare regole di replica solo per i bucket gestiti dal tuo account. Per creare regole di replica per i bucket esterni, saranno i relativi proprietari a dover creare tali regole.

6. Nella pagina Creazione di regole di replica, scegli il modello Replica oggetti da uno o più bucket di origine a uno o più bucket di destinazione.

   > **Important**
   >
   > Quando crei regole di replica utilizzando questo modello, tali regole sostituiscono qualsiasi regola di replica esistente già assegnata al bucket.
   > Per aggiungere o modificare le regole di replica esistenti anziché sostituirle, vai alla scheda Management (Gestione) di ciascun bucket nella console, quindi modifica le regole nella sezione Replication rules (Regole di replica). È inoltre possibile aggiungere

o modificare le regole di replica esistenti utilizzando la AWS CLI, gli SDK o la REST API. Per ulteriori informazioni, consulta Configurazione di replica.

7. Nella sezione Origine e destinazione, in Bucket di origine, seleziona uno o più bucket da cui desideri eseguire la replica degli oggetti. Tutti i bucket (di origine e di destinazione) scelti per la replica devono avere la funzionalità S3 Controllo delle versioni abilitata e ogni bucket deve trovarsi in una Regione AWS diversa. Per ulteriori informazioni sulla funzionalità S3 di controllo delle versioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

   In Bucket di destinazione, seleziona uno o più bucket in cui desideri eseguire la replica degli oggetti.

   > **Note**
   >
   > Assicurati di disporre delle autorizzazioni di lettura e replica necessarie per eseguire la replica; in caso contrario, verranno restituiti errori. Per ulteriori informazioni, consulta Creazione di un ruolo IAM.

8. Nella sezione Replication rule configuration (Configurazione regole di replica), scegli se la regola di replica sarà abilitata (opzione Enabled) o disabilitata (opzione Disabled) al momento della creazione.

   > **Note**
   >
   > Non è possibile immettere un nome nella casella Replication rule name (Nome regola di replica). I nomi delle regole di replica vengono generati in base alla configurazione definita dall'utente quando crea la regola di replica.

9. Nella sezione Scope (Ambito), scegli l'ambito appropriato per la replica.

   - Per replicare l'intero bucket, scegli Apply to all objects in the bucket (Applica a tutti gli oggetti nel bucket).
   - Per replicare un sottoinsieme di oggetti nel bucket, scegli Limit the scope of this rule using one or more filters (Limita l'ambito di questa regola utilizzando uno o più filtri).

     Puoi filtrare gli oggetti utilizzando un prefisso, tag di oggetti o una combinazione di entrambi.

     - Per limitare la replica a tutti gli oggetti con nomi che iniziano con la stessa stringa (ad esempio, `pictures`), immetti un prefisso nella caselal Prefix (Prefisso).

Se specifichi un prefisso corrispondente al nome di una cartella, devi utilizzare un delimitatore, ad esempio / (barra), per indicarne il livello nella gerarchica (ad esempio, `pictures/`). Per ulteriori informazioni sui prefissi, consulta Organizzazione degli oggetti utilizzando i prefissi.

- Per replicare tutti gli oggetti contenenti uno o più tag di oggetto, seleziona Add tag (Aggiungi tag) e specifica la coppia valore-chiave nelle caselle. Per aggiungere un altro tag, ripeti la procedura. Per ulteriori informazioni sui tag degli oggetti, consulta Suddivisione in categorie dello storage utilizzando i tag.

10. Scorri verso il basso fino alla sezione Additional replication options (Opzioni di replica aggiuntive) e seleziona le opzioni di replica che desideri applicare.

> **Note**
>
> Ti consigliamo di applicare le seguenti opzioni:
>
> - Replication time control (RTC) (Controllo tempo di replica [RTC]): per replicare i dati in regioni diverse in un intervallo di tempo prevedibile, puoi utilizzare la funzionalità di controllo del tempo di replica di S3 (S3 RTC). S3 RTC replica il 99,99% dei nuovi oggetti archiviati in Amazon S3 entro 15 minuti, secondo un Accordo sul Livello di Servizio (SLA). Per ulteriori informazioni, consulta the section called “Utilizzo di S3 Replication Time Control”.
> - Replication metrics and notifications (Parametri di replica e notifiche): abilita i parametri di Amazon CloudWatch per monitorare gli eventi di replica.
> - Replica del contrassegno di eliminazione: i contrassegni di eliminazione creati dalle operazioni di eliminazione di S3 verranno replicati. I contrassegni di eliminazione creati dalle regole del ciclo di vita non vengono replicati. Per ulteriori informazioni, consulta Replica dei contrassegni di eliminazione tra i bucket.
>
> Sono previsti costi aggiuntivi per i parametri di replica e le notifiche di S3 RTC e CloudWatch. Per ulteriori dettagli, consulta Prezzi di Amazon S3 e Prezzi di Amazon CloudWatch.

11. Se stai scrivendo una nuova regola di replica che sostituisce una esistente, seleziona I acknowledge that by choosing Create replication rules, these existing replication rules will be

overwritten (Riconosco che scegliendo Crea regole di replica, queste regole di replica esistenti verranno sovrascritte).

12. Scegli Creazione di regole di replica per creare e salvare la nuova regola di replica unidirezionale.

Creare regole di replica bidirezionale per il punto di accesso multi-regione

Le regole di replica consentono la copia asincrona e automatica di oggetti tra bucket. Una regola di replica bidirezionale garantisce che i dati vengano sincronizzati completamente tra due o più bucket in Regioni AWS diverse. Per impostare la replica bidirezionale, viene creata una regola di replica dal bucket di origine (DOC-EXAMPLE-BUCKET-1) al bucket contenente le repliche (DOC-EXAMPLE-BUCKET-2). Quindi, viene creata una seconda regola di replica dal bucket contenente le repliche (DOC-EXAMPLE-BUCKET-2) al bucket di origine (DOC-EXAMPLE-BUCKET-1).

Come tutte le regole di replica, puoi applicare la regola di replica bidirezionale all'intero bucket Amazon S3 o a un sottoinsieme di oggetti filtrati per prefisso o tag di oggetto. Puoi anche mantenere sincronizzate le modifiche dei metadati dei tuoi oggetti abilitando la sincronizzazione delle modifiche delle repliche per ogni regola di replica. Puoi abilitare la sincronizzazione delle modifiche della repliche tramite la console Amazon S3, la AWS CLI, gli SDK AWS, la REST API Amazon S3 o AWS CloudFormation.

Per monitorare l'avanzamento della replica degli oggetti e dei metadati degli oggetti in Amazon CloudWatch, abilita i parametri e le notifiche di Replica S3. Per maggiori informazioni, consulta Monitoraggio dell'avanzamento con i parametri di replica e le notifiche di eventi Amazon S3.

Per creare una regola di replica bidirezionale per il punto di accesso multi-regione

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Multi-Region Access Points (Punti di accesso multi-regione).
3. Scegli il nome del punto di accesso multi-regione da aggiornare.
4. Scegli la scheda Replication and failover (Replica e failover).
5. Scorri verso il basso fino alla sezione Replication rules (Regole di replica) e quindi scegli Create replication rules (Crea regole di replica).

6. Nella pagina Create replication rules (Crea regole di replica), scegli il modello Replicate objects among all specified buckets (Replica oggetti tra tutti i bucket specificati). Il modello Replicate objects among all specified buckets (Replica oggetti tra tutti i bucket specificati) imposta la replica bidirezionale (con funzionalità di failover) per i bucket.

   > **Important**
   >
   > Quando crei regole di replica utilizzando questo modello, tali regole sostituiscono qualsiasi regola di replica esistente già assegnata al bucket.
   > Per aggiungere o modificare le regole di replica esistenti anziché sostituirle, vai alla scheda Management (Gestione) di ciascun bucket nella console, quindi modifica le regole nella sezione Replication rules (Regole di replica). È inoltre possibile aggiungere o modificare le regole di replica esistenti utilizzando la AWS CLI, gli SDK AWS o la REST API Amazon S3. Per ulteriori informazioni, consulta Configurazione di replica.

7. Nella sezione Buckets (Bucket), seleziona almeno due bucket da cui desideri replicare gli oggetti. Tutti i bucket scelti per la replica devono avere la funzionalità S3 di controllo delle versioni abilitata e ogni bucket deve trovarsi in una Regione AWS diversa. Per ulteriori informazioni sulla funzionalità S3 di controllo delle versioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

   > **Note**
   >
   > Assicurati di disporre delle autorizzazioni di lettura e replica necessarie per eseguire la replica; in caso contrario, verranno restituiti errori. Per ulteriori informazioni, consulta Creazione di un ruolo IAM.

8. Nella sezione Replication rule configuration (Configurazione regole di replica), scegli se la regola di replica sarà abilitata (opzione Enabled) o disabilitata (opzione Disabled) al momento della creazione.

   > **Note**
   >
   > Non è possibile immettere un nome nella casella Replication rule name (Nome regola di replica). I nomi delle regole di replica vengono generati in base alla configurazione definita dall'utente quando crea la regola di replica.

9. Nella sezione Scope (Ambito), scegli l'ambito appropriato per la replica.

- Per replicare l'intero bucket, scegli Apply to all objects in the bucket (Applica a tutti gli oggetti nel bucket).
- Per replicare un sottoinsieme di oggetti nel bucket, scegli Limit the scope of this rule using one or more filters (Limita l'ambito di questa regola utilizzando uno o più filtri).

  Puoi filtrare gli oggetti utilizzando un prefisso, tag di oggetti o una combinazione di entrambi.

  - Per limitare la replica a tutti gli oggetti con nomi che iniziano con la stessa stringa (ad esempio, `pictures`), immetti un prefisso nella caselal Prefix (Prefisso).

    Se si immette un prefisso corrispondente al nome di una cartella, è necessario utilizzare `/` (barra) come ultimo carattere (ad esempio, `pictures/`).
  - Per replicare tutti gli oggetti contenenti uno o più tag di oggetto, seleziona Add tag (Aggiungi tag) e specifica la coppia valore-chiave nelle caselle. Per aggiungere un altro tag, ripeti la procedura. Per ulteriori informazioni sui tag degli oggetti, consulta Suddivisione in categorie dello storage utilizzando i tag.

10. Scorri verso il basso fino alla sezione Additional replication options (Opzioni di replica aggiuntive) e seleziona le opzioni di replica che desideri applicare.

> **Note**
>
> Ti consigliamo di applicare le seguenti opzioni, soprattutto se intendi configurare il tuo punto di accesso multi-regione in modo che supporti il failover:
>
> - Replication time control (RTC) (Controllo tempo di replica [RTC]): per replicare i dati in regioni diverse in un intervallo di tempo prevedibile, puoi utilizzare la funzionalità di controllo del tempo di replica di S3 (S3 RTC). S3 RTC replica il 99,99% dei nuovi oggetti archiviati in Amazon S3 entro 15 minuti, secondo un Accordo sul Livello di Servizio (SLA). Per ulteriori informazioni, consulta the section called “Utilizzo di S3 Replication Time Control”.
> - Replication metrics and notifications (Parametri di replica e notifiche): abilita i parametri di Amazon CloudWatch per monitorare gli eventi di replica.
> - Replica del contrassegno di eliminazione: i contrassegni di eliminazione creati dalle operazioni di eliminazione di S3 verranno replicati. I contrassegni di eliminazione creati dalle regole del ciclo di vita non vengono replicati. Per ulteriori informazioni, consulta Replica dei contrassegni di eliminazione tra i bucket.

- Replica modification sync (Sincronizzazione modifiche repliche): abilita la sincronizzazione delle modifiche delle repliche per ogni regola di replica per mantenere sincronizzate anche le modifiche dei metadati degli oggetti. Per ulteriori informazioni, consulta Abilitazione della sincronizzazione delle modifiche alla replica.

Sono previsti costi aggiuntivi per i parametri di replica e le notifiche di S3 RTC e CloudWatch. Per ulteriori dettagli, consulta Prezzi di Amazon S3 e Prezzi di Amazon CloudWatch.

11. Se stai scrivendo una nuova regola di replica che sostituisce una esistente, seleziona I acknowledge that by choosing Create replication rules, these existing replication rules will be overwritten (Riconosco che scegliendo Crea regole di replica, queste regole di replica esistenti verranno sovrascritte).
12. Scegli Create replication rules (Crea regole di replica) per creare e salvare le nuove regole di replica bidirezionale.

Visualizzare regole di replica bidirezionale per il punto di accesso multi-regione

Con i punti di accesso multi-regione, puoi impostare regole di replica unidirezionale o bidirezionale. Per informazioni su come gestire le regole di replica, consulta Gestione delle regole di replica utilizzando la console di Amazon S3.

Per visualizzare regole di replica bidirezionale per il punto di accesso multi-regione

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Multi-Region Access Points (Punti di accesso multi-regione).
3. Scegli il nome del punto di accesso multi-regione.
4. Scegli la scheda Replication and failover (Replica e failover).
5. Scorri verso il basso fino alla sezione Regole di replica. In questa sezione sono elencate tutte le regole di replica create per il punto di accesso multi-regione.

> **Note**
>
> Se al punto di accesso multi-regione corrente hai aggiunto un bucket da un altro account, devi ottenere l'autorizzazione `s3:GetBucketReplication` dal proprietario del bucket per visualizzare le regole di replica per tale bucket.

Utilizzo dei punti di accesso multi-regione con operazioni API supportate

Amazon S3 offre un insieme di operazioni che permettono di gestire i punti di accesso multi-regione. Amazon S3 elabora alcune di queste operazioni in modo sincrono e alcune in modo asincrono. Quando richiami un'operazione asincrona, per prima cosa Amazon S3 autorizza in modo sincrono l'operazione richiesta. Se l'autorizzazione ha esito positivo, Amazon S3 restituisce un token che puoi utilizzare per monitorare lo stato di avanzamento e i risultati dell'operazione richiesta.

> **Note**
>
> Le richieste effettuate tramite la console Amazon S3 sono sempre sincrone. La console attende il completamento della richiesta prima di consentire l'invio di un'altra richiesta.

Puoi visualizzare lo stato e i risultati correnti delle operazioni asincrone utilizzando la console oppure puoi utilizzarli `DescribeMultiRegionAccessPointOperation` negli AWS SDK o nell' AWS CLI API REST. Amazon S3 fornisce un token di tracciamento nella risposta a un'operazione asincrona. Includi quel token di tracciamento come argomento per `DescribeMultiRegionAccessPointOperation`. Quando includi il token di monitoraggio, Amazon S3 restituisce lo stato corrente e i risultati dell'operazione specificata, inclusi eventuali errori o informazioni pertinenti sulla risorsa. Amazon S3 esegue le operazioni `DescribeMultiRegionAccessPointOperation` in modo sincrono.

Tutte le richieste del piano di controllo (control-plane) per creare o mantenere punti di accesso multi-regione devono essere instradate alla regione `US West (Oregon)`. Per richieste sul piano dati del punto di accesso multi-regione, non è necessario specificare le regioni. Per il piano di controllo (control-plane) di failover del punto di accesso multi-regione, la richiesta deve essere instradata a una delle cinque regioni supportate. Per ulteriori informazioni sulle regioni supportate da punti di accesso multiregionali, consulta. Restrizioni e limitazioni dei punti di accesso multi-regione

Inoltre, è necessario concedere l'`s3:ListAllMyBuckets`autorizzazione all'utente, al ruolo o a un'altra entità AWS Identity and Access Management (IAM) che effettua una richiesta per gestire un punto di accesso multiregionale.

Negli esempi seguenti viene illustrato come utilizzare i punti di accesso multi-regione con operazioni compatibili in Amazon S3.

Argomenti

Compatibilità con punti di accesso multiregionali e SDK Servizi AWSAWS

Per utilizzare un punto di accesso multiregionale con applicazioni che richiedono un nome bucket Amazon S3, utilizza l'Amazon Resource Name (ARN) del punto di accesso multiregionale quando effettui richieste utilizzando un SDK. AWS Per verificare quali AWS SDK sono compatibili con i punti di accesso multiregionali, consulta Compatibilità con gli SDK. AWS

Compatibilità dei punti di accesso multi-regione con le operazioni S3

Puoi utilizzare le seguenti operazioni API del piano dati Amazon S3 per eseguire azioni sugli oggetti nei bucket associati al punto di accesso multi-regione. Le seguenti operazioni S3 possono accettare ARN di punti di accesso multi-regione:

- AbortMultipartUpload
- CompleteMultipartUpload

- CreateMultipartUpload
- DeleteObject
- DeleteObjectTagging
- GetObject
- GetObjectAcl
- GetObjectLegalHold
- GetObjectRetention
- GetObjectTagging
- HeadObject
- ListMultipartUploads
- ListObjectsV2
- ListParts
- PutObject
- PutObjectAcl
- PutObjectLegalHold
- PutObjectRetention
- PutObjectTagging
- RestoreObject
- UploadPart

> **Note**
>
> I punti di accesso multiregione supportano le operazioni di copia utilizzando punti di accesso multiregione solo come destinazione quando si utilizza l'ARN del punto di accesso multiregionale.

Puoi utilizzare le seguenti operazioni del piano di controllo (control-plane) Amazon S3 per creare e gestire i punti di accesso multi-regione:

- CreateMultiRegionAccessPoint
- DescribeMultiRegionAccessPointOperation

- GetMultiRegionAccessPoint
- GetMultiRegionAccessPointPolicy
- GetMultiRegionAccessPointPolicyStatus
- GetMultiRegionAccessPointRoutes
- ListMultiRegionAccessPoints
- PutMultiRegionAccessPointPolicy
- SubmitMultiRegionAccessPointRoutes

Visualizzare la configurazione di instradamento del punto di accesso multi-regione

AWS CLI

Il seguente comando di esempio recupera la configurazione di instradamento del punto di accesso multi-regione in modo da poter visualizzare gli stati di instradamento correnti per i bucket. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
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

Il seguente codice SDK per Java recupera la configurazione di instradamento del punto di accesso multi-regione in modo da poter visualizzare gli stati di instradamento correnti per i bucket. Per utilizzare questa sintassi di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
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

Il seguente SDK per il JavaScript codice recupera la configurazione del percorso del punto di accesso multiregionale in modo da poter visualizzare gli stati di routing correnti per i bucket. Per utilizzare questa sintassi di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
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

Il seguente codice SDK per Python recupera la configurazione di instradamento del punto di accesso multi-regione in modo da poter visualizzare gli stati di instradamento correnti per i bucket. Per utilizzare questa sintassi di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
s3.get_multi_region_access_point_routes(
        AccountId=111122223333,
        Mrap=arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap)['Routes']
```

Aggiornare la policy di bucket Amazon S3 sottostante

Per garantire un accesso adeguato, devi anche aggiornare la policy di bucket Amazon S3 sottostante. Nei seguenti esempi il controllo dell'accesso viene delegato alla policy del punto di accesso multi-regione. Dopo aver delegato il controllo dell'accesso alla policy del punto di accesso multi-regione, la policy del bucket non viene più utilizzata per il controllo dell'accesso quando le richieste vengono effettuate tramite il punto di accesso multi-regione.

Di seguito è riportato un esempio di policy di bucket che delega il controllo degli accessi alla policy del punto di accesso multi-regione. Per utilizzare questa policy di bucket, sostituisci *`user input placeholders`* con le tue informazioni. Per applicare questo criterio tramite il AWS CLI `put-bucket-policy` comando, come illustrato nell'esempio successivo, salvate il criterio in un file, ad esempio. `policy.json`

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Principal": { "AWS": "*" },
    "Effect": "Allow",
    "Action": ["s3:*"],
    "Resource": ["arn:aws:s3:::111122223333/*", "arn:aws:s3:::DOC-EXAMPLE-BUCKET"],
    "Condition": {
      "StringEquals": {
        "s3:DataAccessPointAccount": "444455556666"
      }
    }
  }
}
```

Il seguente comando di esempio `put-bucket-policy` associa la policy del bucket S3 aggiornata al bucket S3:

```
aws s3api put-bucket-policy
  --bucket DOC-EXAMPLE-BUCKET
  --policy file:///tmp/policy.json
```

Aggiornare la configurazione di instradamento di un punto di accesso multi-regione

Il seguente comando di esempio aggiorna la configurazione di instradamento del punto di accesso multi-regione. I comandi di instradamento del punto di accesso multi-regione possono essere eseguiti nelle seguenti cinque regioni:

- `ap-southeast-2`
- `ap-northeast-1`
- `us-east-1`
- `us-west-2`
- `eu-west-1`

In una configurazione di instradamento dei punti di accesso multi-regione, è possibile impostare i bucket su uno stato di instradamento attivo o passivo. A differenza dei bucket passivi, i bucket attivi ricevono traffico. È possibile impostare lo stato di instradamento di un bucket impostando il valore `TrafficDialPercentage` del bucket su `100` per attivo o su `0` per passivo.

AWS CLI

Il seguente comando di esempio aggiorna la configurazione di instradamento per i punti di accesso multi-regione. In questo esempio, *DOC-EXAMPLE-BUCKET1* è impostato sullo stato attivo e *DOC-EXAMPLE-BUCKET2* su passivo. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
aws s3control submit-multi-region-access-point-routes
--region ap-southeast-2
--account-id 111122223333
--mrap arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap
--route-updates Bucket=DOC-EXAMPLE-BUCKET1,TrafficDialPercentage=100
                Bucket=DOC-EXAMPLE-BUCKET2,TrafficDialPercentage=0
```

SDK for Java

Il seguente codice SDK per Java aggiorna la configurazione di instradamento del punto di accesso multi-regione. Per utilizzare questa sintassi di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
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

Il seguente SDK per il JavaScript codice aggiorna la configurazione del percorso del punto di accesso multiregionale. Per utilizzare questa sintassi di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
const REGION = 'ap-southeast-2'

const s3ControlClient = new S3ControlClient({
  region: REGION
})

export const run = async () => {
  try {
    const data = await s3ControlClient.send(
      new SubmitMultiRegionAccessPointRoutesCommand({
        AccountId: '111122223333',
        Mrap: 'arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap',
        RouteUpdates: [
```

```
          {
            Region: 'eu-west-1',
            TrafficDialPercentage: 100,
          },
          {
            Region: 'ca-central-1',
            Bucket: 'DOC-EXAMPLE-BUCKET1',
            TrafficDialPercentage: 0,
          },
        ],
      })
    )
    console.log('Success', data)
    return data
  } catch (err) {
    console.log('Error', err)
  }
}

run()
```

SDK for Python

Il seguente codice SDK per Python aggiorna la configurazione di instradamento del punto di accesso multi-regione. Per utilizzare questa sintassi di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
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
        }])
```

Aggiunta di un oggetto a un bucket nel punto di accesso multi-regione

Per aggiungere un oggetto al bucket associato al punto di accesso multi-regione, puoi utilizzare l'operazione PutObject. Per mantenere sincronizzati tutti i bucket nel punto di accesso multi-regione, abilita Replica tra regioni.

> **Note**
>
> Per utilizzare questa operazione, devi disporre dell'autorizzazione `s3:PutObject` per il punto di accesso multi-regione. Per ulteriori informazioni sui requisiti di autorizzazione del punto di accesso multi-regione, consultare Autorizzazioni.

AWS CLI

La seguente richiesta del piano dati di esempio carica *example.txt* nel punto di accesso multi-regione specificato. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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!'
)
```

Recupero degli oggetti dal punto di accesso multi-regione

Per recuperare oggetti dal punto di accesso multi-regione, puoi utilizzare l'operazione GetObject.

> **Note**
>
> Per utilizzare questa operazione API, devi disporre dell'autorizzazione `s3:GetObject` per il punto di accesso multi-regione. Per ulteriori informazioni sui requisiti di autorizzazione del punto di accesso multi-regione, consultare Autorizzazioni.

AWS CLI

La seguente richiesta del piano dati di esempio recupera *example.txt* dal punto di accesso multi-regione specificato e lo scarica come *downloaded_example.txt*. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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'
)
```

Elencare gli oggetti archiviati in un bucket sottostante il punto di accesso multi-regione

Per restituire un elenco di oggetti archiviati in un bucket sottostante il punto di accesso multi-regione, utilizza l'operazione ListObjectsV2. Nel comando di esempio seguente, tutti gli oggetti per il punto di accesso multi-regione specificato vengono elencati utilizzando l'ARN per il punto di accesso multi-regione. In questo caso, l'ARN del punto di accesso multi-regione è:

`arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap`

> **Note**
>
> Per utilizzare questa operazione API, devi disporre dell'autorizzazione `s3:ListBucket` per il punto di accesso multi-regione e il bucket sottostante. Per ulteriori informazioni sui requisiti di autorizzazione del punto di accesso multi-regione, consultare Autorizzazioni.

AWS CLI

La seguente richiesta del piano dati di esempio elenca gli oggetti nel bucket che sta alla base del punto di accesso multi-regione specificato dall'ARN. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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'
)
```

Utilizzare un URL prefirmato con i punti di accesso multi-regione

Puoi utilizzare un URL prefirmato per generare un URL che consenta ad altri utenti di accedere ai bucket Amazon S3 tramite un punto di accesso multi-regione di Amazon S3. Quando crei un URL prefirmato, associalo a un'operazione sugli oggetti specifica come un caricamento S3 (`PutObject`) o un download S3 (`GetObject`). L'URL prefirmato può essere condiviso e gli utenti che dispongono dell'autorizzazione di accesso possono eseguire l'azione incorporata nell'URL come se fossero l'utente di firma originale.

Gli URL prefirmati hanno una data di scadenza. Una volta raggiunta la scadenza, l'URL non funzionerà più.

Prima di utilizzare i punti di accesso multi-regione S3 con URL prefirmati, verifica la compatibilità dell'SDK AWS con l'algoritmo SigV4A. Verifica che la tua versione SDK supporti SigV4a come

implementazione di firma utilizzata per firmare le richieste globali a livello di Regione AWS . Per ulteriori informazioni sugli URL prefirmati, consulta Condivisione di oggetti mediante URL prefirmati.

Negli esempi riportati di seguito viene illustrato come utilizzare i punti di accesso multi-regione con URL prefirmati. Per usare questi esempi, sostituisci *user input placeholders* con le tue informazioni.

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**
>
> Per utilizzare SigV4a con credenziali di sicurezza temporanee, ad esempio quando utilizzi ruoli IAM, assicurati di richiedere le credenziali temporanee a un endpoint regionale in (), anziché a un endpoint globale. AWS Security Token Service AWS STS Se utilizzi l'endpoint globale for AWS STS (`sts.amazonaws.com`), AWS STS genererà credenziali temporanee da un endpoint globale, che non è supportato da Sig4A. Di conseguenza, verrà restituito un errore. Per risolvere questo problema, utilizza uno degli endpoint regionali elencati per. AWS STS

Utilizzare un bucket configurato con l'opzione di pagamento a carico del richiedente con i punti di accesso multi-regione

Se un bucket S3 associato ai punti di accesso multi-regione è configurato per utilizzare l'opzione Pagamento a carico del richiedente, il richiedente pagherà la richiesta di creazione del bucket, il download e gli eventuali costi relativi ai punti di accesso multi-regione. Per ulteriori informazioni, consulta i Prezzi di Amazon S3.

Di seguito è riportato un esempio di richiesta di piano dati a un punto di accesso multi-regione connesso a un bucket con pagamento a carico del richiedente.

AWS CLI

Per scaricare oggetti da un punto di accesso multi-regione collegato a un bucket con pagamento a carico del richiedente, devi specificare `--request-payer requester` come parte della richiesta get-object. Inoltre, devi specificare il nome del file nel bucket e la posizione in cui archiviare il file scaricato.

```
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

Per scaricare oggetti da un punto di accesso multi-regione collegato a un bucket con pagamento a carico del richiedente, devi specificare `RequestPayer.REQUESTER` come parte della richiesta

`GetObject`. È inoltre necessario specificare il nome del file nel bucket e la posizione in cui deve essere archiviato.

```
GetObjectResponse getObjectResponse = s3Client.getObject(GetObjectRequest.builder()
    .key("example-file.txt")
    .bucket("arn:aws:s3::
123456789012:accesspoint/abcdef0123456.mrap")
    .requestPayer(RequestPayer.REQUESTER)
    .build()
).response();
```

Monitoraggio e registrazione delle richieste effettuate tramite un punto di accesso multi-regione alle risorse sottostanti

Amazon S3 registra le richieste effettuate tramite i punti di accesso multi-regione e le richieste effettuate alle operazioni API che li gestiscono, ad esempio `CreateMultiRegionAccessPoint` e `GetMultiRegionAccessPointPolicy`. Le richieste effettuate ad Amazon S3 tramite un punto di accesso multi-regione vengono visualizzate nei registri di accesso del server Amazon S3 e nei registri AWS CloudTrail con il nome host del punto di accesso multi-regione. Il nome host di un punto di accesso ha il formato *`MRAP_alias`*`.accesspoint.s3-global.amazonaws.com`. Ad esempio, supponiamo di disporre della seguente configurazione di bucket e punto di accesso multi-regione:

- Un bucket denominato `my-bucket-usw2` nella regione `us-west-2` che contiene l'oggetto `my-image.jpg`
- Un bucket denominato `my-bucket-aps1` nella regione `ap-south-1` che contiene l'oggetto `my-image.jpg`
- Un bucket denominato `my-bucket-euc1` nella regione `eu-central-1` che non contiene un oggetto denominato `my-image.jpg`.
- Un punto di accesso multi-regione denominato `my-mrap` con l'alias `mfzwi23gnjvgw.mrap`configurato per soddisfare le richieste da tutti e tre i bucket.
- L'ID dell'account AWS è `123456789012`.

Una richiesta eseguita per recuperare `my-image.jpg` direttamente attraverso uno qualsiasi dei bucket appare nei registri con il nome host *`bucket_name`*`.s3.`*`Region`*`.amazonaws.com`.

Se invece esegui la richiesta tramite il punto di accesso multi-regione, Amazon S3 determina innanzitutto quali bucket nelle diverse regioni si trovano più vicini. Dopo che Amazon S3 determina i bucket utilizzare per eseguire la richiesta, invia la richiesta a tale bucket e registra l'operazione utilizzando il nome host del punto di accesso multi-regione. In questo esempio, se Amazon S3 inoltra la richiesta a `my-bucket-aps1`, i tuoi log riporteranno una richiesta `GET` riuscita per `my-image.jpg` da `my-bucket-aps1`, utilizzando `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com` come nome host.

> **Important**
>
> I punti di accesso multi-regione non rilevano il contenuto dei dati dei bucket sottostanti. Pertanto, il bucket che riceve la richiesta potrebbe non contenere i dati richiesti. Se Amazon S3 determina che il bucket `my-bucket-euc1` è il più vicino, i log includeranno una richiesta `GET` non riuscita per `my-image.jpg` da `my-bucket-euc1`, utilizzando `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com` come nome host. Se la richiesta è stata invece instradata a `my-bucket-usw2`, i tuoi log indicherebbero una richiesta `GET` riuscita.

Per ulteriori informazioni sui log degli accessi al server Amazon S3, consulta Registrazione delle richieste con registrazione dell'accesso al server. Per ulteriori informazioni su AWS CloudTrail, consulta Che cos'è AWS CloudTrail? nella Guida per l'utente di AWS CloudTrail.

Monitoraggio e registrazione delle richieste effettuate alle operazioni API di gestione dei punti di accesso multi-regione

Amazon S3 fornisce diverse operazioni API per gestire i punti di accesso multi-regione, come `CreateMultiRegionAccessPoint` e `GetMultiRegionAccessPointPolicy`. Quando inoltri richieste a queste operazioni API mediante la AWS Command Line Interface (AWS CLI), gli SDK AWS o la REST API Amazon S3, Amazon S3 elabora queste richieste in modo asincrono. Se disponi delle autorizzazioni appropriate per la richiesta, Amazon S3 restituisce un token per queste richieste. Puoi usare questo token con `DescribeAsyncOperation` per semplificare la visualizzazione dello stato delle operazioni asincrone in corso. Amazon S3 elabora le richieste `DescribeAsyncOperation` in modo sincrono. Per visualizzare lo stato delle richieste asincrone, puoi utilizzare la console Amazon S3, la AWS CLI, gli SDK o la REST API.

> **Note**
>
> La console visualizza solo lo stato delle richieste asincrone effettuate nei 14 giorni precedenti. Per visualizzare lo stato delle richieste meno recenti, utilizza AWS CLI, SDK o l'API REST.

Le operazioni di gestione asincrona possono avere uno tra diversi stati:

NEW

Amazon S3 ha ricevuto la richiesta e si sta preparando per eseguire l'operazione.

IN_PROGRESS

Amazon S3 sta attualmente eseguendo l'operazione.

SUCCESS

L'operazione è stata completata. La risposta include informazioni rilevanti, ad esempio l'alias del punto di accesso multi-regione per una richiesta `CreateMultiRegionAccessPoint`.

FAILED

L'operazione ha avuto esito negativo. La risposta include un messaggio di errore che indica il motivo dell'errore.

Utilizzo di AWS CloudTrail con i punti di accesso multi-regione

Puoi utilizzare AWS CloudTrail per visualizzare, cercare, scaricare, archiviare, analizzare e rispondere alle attività dell'account all'interno dell'infrastruttura AWS. Con i punti di accesso multi-regione e la registrazione CloudTrail, puoi recuperare le seguenti informazioni:

- Chi o cosa ha eseguito l'operazione e l'operazione eseguita
- Le risorse utilizzate
- Quando si è verificato l'evento
- Altri dettagli relativi all'evento

Puoi utilizzare queste informazioni di registrazione per aiutarti ad analizzare e rispondere alle attività che si sono verificate nei punti di accesso multi-regione.

Come configurare AWS CloudTrail per i punti di accesso multi-regione

Per abilitare la registrazione CloudTrail per qualsiasi operazione relativa alla creazione o alla gestione dei punti di accesso multi-regione, devi configurarla in modo che registri gli eventi nella regione Stati Uniti occidentali (Oregon). Devi impostare la configurazione della registrazione in questo modo indipendentemente dalla regione in cui ti trovi quando esegui la richiesta o dalle regioni supportate dal punto di accesso multi-regione. Tutte le richieste di creazione o gestione di un punto di accesso multi-regione vengono instradate attraverso la regione Stati Uniti occidentali (Oregon). Ti consigliamo di aggiungere questa regione a un trail esistente o crearne uno nuovo che contenga questa regione e tutte le regioni associate al punto di accesso multi-regione.

Amazon S3 registra le richieste eseguite tramite un punto di accesso multi-regione e quelle eseguite alle operazioni API che gestiscono i punti di accesso, ad esempio `CreateMultiRegionAccessPoint` e `GetMultiRegionAccessPointPolicy`. Quando registri queste richieste tramite un punto di accesso multi-regione, queste vengono visualizzate nei registri AWS CloudTrail con il nome host del punto di accesso multi-regione. Ad esempio, se effettui richieste a un bucket tramite un punto di accesso multi-regione con l'alias `mfzwi23gnjvgw.mrap`, le voci nel log di CloudTrail hanno il nome host `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`.

I punti di accesso multi-regione instradano le richieste al bucket più vicino. Per questo motivo, quando esamini i log di CloudTrail per un punto di accesso multi-regione, verranno visualizzate le richieste effettuate ai bucket sottostanti. Alcune di queste richieste potrebbero corrispondere a richieste dirette al bucket non instradate attraverso il punto di accesso multi-regione. Considera questa eventualità quando esamini il traffico. Quando un bucket si trova in un punto di accesso multi-regione, è comunque possibile effettuare richieste direttamente a tale bucket senza passare attraverso il punto di accesso multi-regione.

Esistono eventi asincroni coinvolti nella creazione e nella gestione dei punti di accesso multi-regione. Le richieste asincrone non hanno eventi di completamento nel registro di CloudTrail. Per ulteriori informazioni sul monitoraggio delle richieste asincrone, consulta Monitoraggio e registrazione delle richieste effettuate alle operazioni API di gestione dei punti di accesso multi-regione.

Per ulteriori informazioni su AWS CloudTrail, consulta Che cos'è AWS CloudTrail? nella Guida per l'utente di AWS CloudTrail.

Sicurezza di Amazon S3

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il modello di responsabilità condivisa descrive questo come sicurezza del cloud e sicurezza nel cloud:

Sicurezza del cloud

AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi in Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. L'efficacia della nostra sicurezza è regolarmente testata e verificata da revisori di terze parti come parte dei programmi di conformitàAWS. Per ulteriori informazioni sui programmi di conformità che si applicano ad Amazon S3, consulta AWS Servizi coperti dal programma di conformità.

Sicurezza nel cloud

La tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è anche responsabile per altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda e leggi e normative applicabili. Per Amazon S3, la tua responsabilità include le seguenti aree:

- Gestione dei dati, inclusa la proprietà degli oggetti e la crittografia.
- Classificazione delle tue risorse.
- Gestione degli accessi ai tuoi dati tramite ruoli IAM e altre configurazioni di servizio per applicare le autorizzazioni appropriate.
- Attivazione di controlli AWS CloudTrailinvestigativi come Amazon GuardDuty per Amazon S3.

Questa documentazione consente di comprendere come applicare il modello di responsabilità condivisa quando utilizzi Amazon S3. Gli argomenti seguenti descrivono come configurare Amazon S3 per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche come utilizzare altri AWS servizi che possono aiutarti a monitorare e proteggere le tue risorse Amazon S3.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Argomenti

Protezione dei dati in Amazon S3

Amazon S3 offre un'infrastruttura di storage estremamente durevole, concepita per lo storage dei dati mission-critical e primari. S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive archiviano in modo ridondante gli oggetti su più dispositivi in un minimo di tre zone di disponibilità in un Regione AWS. Una zona di disponibilità consiste in uno o più data center separati con alimentazione, rete e connettività ridondanti in una Regione AWS. Le zone di disponibilità sono fisicamente separate da una distanza significativa, di diversi chilometri, da qualsiasi altra zona di disponibilità, anche se tutte si trovano nel raggio di 100 km (60 miglia) l'una dall'altra. La classe di archiviazione S3 One Zone — IA consente di archiviare i dati in modo ridondante su più dispositivi all'interno di una singola zona di disponibilità. Questi servizi sono progettati per far fronte ai guasti simultanei dei dispositivi rilevando e riparando

rapidamente eventuali perdite di ridondanza e controllano regolarmente l'integrità dei dati utilizzando checksum.

Lo storage standard Amazon S3 offre le seguenti caratteristiche:

- Sostenuto dall'Accordo sul livello di servizio (SLA) Amazon S3.
- È progettato per garantire una durabilità pari al 99.999999999% e una disponibilità degli oggetti pari al 99.99% per un determinato anno.
- S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive sono tutti progettati per conservare i dati in caso di perdita di un'intera zona di disponibilità Amazon S3.

Amazon S3 protegge ulteriormente i dati tramite la funzione Controllo delle versioni. che può essere impiegata per conservare, recuperare e ripristinare qualsiasi versione di ogni oggetto archiviato nel bucket Amazon S3. Con la funzione Controllo delle versioni si può facilmente eseguire il ripristino dopo errori dell'applicazione e operazioni non intenzionali dell'utente. Per default, le richieste recuperano la versione più recente scritta. È comunque possibile recuperare versioni meno recenti di un oggetto specificandone la versione in una richiesta.

Oltre alla funzionalità S3 di controllo delle versioni, puoi anche utilizzare funzionalità Blocco dell'accesso pubblico Amazon S3 e Replica S3 per proteggere i tuoi dati. Per ulteriori informazioni, consulta Tutorial: Protezione dei dati su Amazon S3 da eliminazioni accidentali o bug delle applicazioni mediante le funzionalità S3 di controllo delle versioni, blocco degli oggetti e replica.

Ai fini della protezione dei dati, ti consigliamo di proteggere Account AWS le credenziali e di configurare account utente individuali con AWS Identity and Access Management, in modo che a ciascun utente vengano concesse solo le autorizzazioni necessarie per svolgere le proprie mansioni lavorative.

Se hai bisogno di moduli crittografici convalidati FIPS 140-2 per l'accesso AWS tramite un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-2.

Le best practice di sicurezza seguenti gestiscono anche la protezione dei dati in 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

Protezione dei dati con la crittografia

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva negli SDK and. AWS Command Line Interface AWS Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

La protezione dei dati ha lo scopo di proteggere i dati sia in transito (durante la trasmissione verso e da Amazon S3), sia quando sono a riposo (ovvero quando sono archiviati su disco nei data center Amazon S3). È possibile proteggere i dati in transito utilizzando Secure Socket Layer/Transport Layer Security (SSL/TLS) o la crittografia lato client. Per la protezione dei dati a riposo in Amazon S3 sono disponibili le opzioni seguenti:

- Crittografia lato server: Amazon S3 crittografa gli oggetti prima di salvarli su dischi AWS nei data center e quindi decrittografa gli oggetti quando li scarichi.

  Tutti i bucket Amazon S3 hanno la crittografia configurata per impostazione predefinita e tutti i nuovi oggetti caricati in un bucket S3 vengono automaticamente crittografati quando sono a riposo. La crittografia lato server con le chiavi gestite da Amazon S3 (SSE-S3) è la configurazione predefinita della crittografia per ogni bucket di Amazon S3. Per utilizzare un diverso tipo di crittografia, puoi specificare il tipo di crittografia lato server da utilizzare nelle richieste PUT S3 oppure impostare la configurazione di crittografia predefinita nel bucket di destinazione.

  Se desideri specificare un tipo di crittografia diverso nelle tue PUT richieste, puoi utilizzare la crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS), la crittografia lato server a due livelli con AWS KMS chiavi (DSSE-KMS) o la crittografia lato server

con chiavi fornite dal cliente (SSE-C). Per impostare una configurazione di crittografia predefinita diversa nel bucket di destinazione puoi utilizzare SSE-KMS o DSSE-KMS.

Per ulteriori informazioni su ogni opzione della crittografia lato server, consulta Protezione dei dati con la crittografia lato server.

Per configurare la crittografia lato server, consulta:

- Specifica della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3)
- Specifica della crittografia lato server con AWS KMS (SSE-KMS)
- the section called “Specifica di DSSE-KMS”
- Specifica della crittografia lato server con chiavi fornite dal cliente (SSE-C)

- Crittografia lato client: esegui la crittografia dei dati sul lato client e carica i dati crittografati in Amazon S3. In questo caso, è l'utente a gestire il processo di crittografia, nonché le chiavi e gli strumenti correlati.

  Per configurare la crittografia lato client, vedi Protezione dei dati con la crittografia lato client.

Per vedere quale percentuale di byte di archiviazione crittografati, puoi utilizzare i parametri di Amazon S3 Storage Lens. S3 Storage Lens è una funzionalità di analisi dell'archiviazione su cloud che puoi utilizzare per avere una panoramica completa a livello di organizzazione sull'utilizzo e sulle attività relative all'archiviazione di oggetti. Per ulteriori informazioni, consulta Valutazione dell'attività e dell'utilizzo dello storage con S3 Storage Lens. Per un elenco completo dei parametri, consulta Glossario dei parametri di S3 Storage.

Per ulteriori informazioni sulla crittografia lato server e sulla crittografia lato client, consulta gli argomenti elencati di seguito.

Argomenti

- Protezione dei dati con la crittografia lato server
- Protezione dei dati con la crittografia lato client

Protezione dei dati con la crittografia lato server

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva negli SDK and. AWS Command Line Interface AWS Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

La crittografia lato server è la crittografia dei dati nella posizione di destinazione eseguita dall'applicazione o dal servizio che li riceve. Amazon S3 crittografa i tuoi dati a livello di oggetto mentre li scrive su dischi nei data AWS center e li decrittografa per te quando ti accedi. Se la richiesta è autenticata e sono disponibili le autorizzazioni per l'accesso, non c'è differenza nelle modalità di accesso agli oggetti, crittografati o meno. Ad esempio, se si condividono gli oggetti tramite un URL prefirmato, quest'ultimo funziona nello stesso modo, sia per i dati crittografati che per quelli non crittografati. Inoltre, quando si richiede un elenco degli oggetti nel bucket, le operazioni API restituisce l'elenco di tutti gli oggetti, crittografati o meno.

Tutti i bucket Amazon S3 hanno la crittografia configurata per impostazione predefinita e tutti i nuovi oggetti caricati in un bucket S3 vengono automaticamente crittografati quando sono a riposo. La crittografia lato server con le chiavi gestite da Amazon S3 (SSE-S3) è la configurazione predefinita della crittografia per ogni bucket di Amazon S3. Per utilizzare un diverso tipo di crittografia, puoi specificare il tipo di crittografia lato server da utilizzare nelle richieste PUT S3 oppure impostare la configurazione di crittografia predefinita nel bucket di destinazione.

Se desideri specificare un tipo di crittografia diverso nelle tue PUT richieste, puoi utilizzare la crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS), la crittografia lato server a due livelli con AWS KMS chiavi (DSSE-KMS) o la crittografia lato server con chiavi fornite dal cliente (SSE-C). Per impostare una configurazione di crittografia predefinita diversa nel bucket di destinazione puoi utilizzare SSE-KMS o DSSE-KMS.

> **Note**
>
> Non è possibile applicare contemporaneamente tipi diversi di crittografia lato server a uno stesso oggetto.

Se devi crittografare gli oggetti esistenti, usa Operazioni in batch S3 e S3 Inventory. Per ulteriori informazioni, consulta Crittografia di oggetti con Operazioni in batch Amazon S3 e Esecuzione di operazioni in batch su larga scala su oggetti Amazon S3.

A seconda di come si sceglie di gestire le chiavi di crittografia e il numero di livelli di crittografia da applicare, sono disponibili quattro opzioni che si escludono a vicenda per la crittografia lato server.

Crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3)

Tutti i bucket Amazon S3 hanno la crittografia configurata per impostazione predefinita. L'opzione predefinita per la crittografia lato server prevede le chiavi gestite da Amazon S3 (SSE-S3). Ogni oggetto è crittografato con una chiave univoca. Come ulteriore tutela, SSE-S3 esegue la crittografia della chiave con una chiave root che ruota con regolarità. Per crittografare i dati, SSE-S3 utilizza una delle cifrature di blocco più complesse disponibili, lo standard di crittografia avanzata a 256 bit (AES-256). Per ulteriori informazioni, consulta Uso della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).

Crittografia lato server con chiavi () (SSE-KMS) AWS Key Management ServiceAWS KMS

La crittografia lato server con AWS KMS keys (SSE-KMS) viene fornita tramite l'integrazione del servizio con AWS KMS Amazon S3. Con AWS KMS, hai un maggiore controllo sulle tue chiavi. Ad esempio, puoi visualizzare le chiavi separate, modificare le policy di controllo e seguire le chiavi in AWS CloudTrail. Inoltre, puoi creare e gestire chiavi gestite dal cliente oppure utilizzare chiavi Chiavi gestite da AWS create appositamente per te, il tuo servizio e la tua regione. Per ulteriori informazioni, consulta Utilizzo della crittografia lato server con chiavi (SSE-KMS) AWS KMS.

Crittografia lato server a doppio livello con AWS Key Management Service (AWS KMS) chiavi (DSSE-KMS)

La crittografia lato server a doppio livello con AWS KMS keys (DSSE-KMS) è simile a SSE-KMS, ma DSSE-KMS applica due singoli livelli di crittografia a livello di oggetto anziché un livello. Poiché entrambi i livelli di crittografia vengono applicati a un oggetto sul lato server, è possibile utilizzare un'ampia gamma di strumenti per analizzare i dati in S3 utilizzando un metodo di crittografia in grado

di Servizi AWS soddisfare i requisiti di conformità. Per ulteriori informazioni, consulta Utilizzo della crittografia lato server a doppio livello con chiavi (DSSE-KMS) AWS KMS.

Crittografia lato server con chiavi fornite dal cliente (SSE-C)

Con la crittografia lato server con chiavi fornite dal cliente (SSE-C) gestisci le chiavi di crittografia, mentre Amazon S3 si occupa di crittografare gli oggetti durante la scrittura su disco e di decrittarli al momento dell'accesso. Per ulteriori informazioni, consulta Utilizzo della crittografia lato server con chiavi fornite dal cliente (SSE-C).

Amazon S3 ora esegue la crittografia automatica di tutti i nuovi oggetti

Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. SSE-S3, che utilizza l'algoritmo Advanced Encryption Standard (AES-256) a 256 bit, viene applicato automaticamente a tutti i nuovi bucket e a qualsiasi bucket S3 esistente per il quale non sia già stata configurata la crittografia predefinita. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva in () e negli SDK. AWS Command Line Interface AWS CLI AWS

Nelle sezioni seguenti vengono fornite le risposte alle domande su questo aggiornamento.

Amazon S3 modifica le impostazioni della crittografia predefinita per i miei bucket esistenti che hanno già configurato la crittografia predefinita?

No. Non sono state apportate modifiche alla configurazione di crittografia predefinita per un bucket esistente che ha già configurato la crittografia SSE-S3 o lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS). Per ulteriori informazioni su come configurare il comportamento della crittografia predefinita per i bucket, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3. Per ulteriori informazioni sulle impostazioni della crittografia SSE-S3 e SSE-KMS, consulta Protezione dei dati con la crittografia lato server.

La crittografia predefinita è abilitata nei miei bucket esistenti che non hanno la crittografia predefinita configurata?

Sì. Amazon S3 ora configura la crittografia predefinita in tutti i bucket non crittografati esistenti per applicare la crittografia lato server con chiavi gestite da S3 come livello base di crittografia per i nuovi

oggetti caricati in questi bucket. Gli oggetti già presenti in un bucket non crittografato esistente non verranno crittografati automaticamente.

Come posso visualizzare lo stato della crittografia predefinita dei caricamenti di nuovi oggetti?

Attualmente, puoi visualizzare lo stato di crittografia predefinito dei caricamenti di nuovi oggetti nei AWS CloudTrail log, S3 Inventory e S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva negli () e negli SDK. AWS Command Line Interface AWS CLI AWS

- Per visualizzare i tuoi CloudTrail eventi, consulta Visualizzazione CloudTrail degli eventi nella console nella Guida per l'utente. CloudTrail AWS CloudTrail CloudTrail i log forniscono il monitoraggio delle API `PUT` e le `POST` richieste ad Amazon S3. Quando viene utilizzata la crittografia predefinita per crittografare gli oggetti nei bucket, i CloudTrail log `PUT` e le richieste `POST` API includeranno il seguente campo come coppia nome-valore:. `"SSEApplied":"Default_SSE_S3"`
- Per visualizzare lo stato di crittografia automatica dei nuovi caricamenti di oggetti in S3 Inventory, configura un report di S3 Inventory che includa il campo dei metadati Encryption (Crittografia), quindi visualizza lo stato di crittografia di ogni nuovo oggetto nel report. Per ulteriori informazioni, consulta Impostazione di Amazon S3 Inventory.
- Per visualizzare lo stato di crittografia automatica per i nuovi caricamenti di oggetti in S3 Storage Lens, configura un pannello di controllo di S3 Storage Lens e visualizza le metriche Encrypted bytes (Byte crittografati) e Encrypted object count (Conteggio degli oggetti crittografati) nella categoria Data protection (Protezione dei dati) del pannello di controllo. Per ulteriori informazioni, consulta Creazione di un pannello di controllo di Amazon S3 Storage Lens e Visualizzazione dei parametri di S3 Storage Lens nei pannelli di controllo.
- Per visualizzare lo stato della crittografia automatica a livello di bucket nella console Amazon S3, controlla la crittografia predefinita dei bucket Amazon S3 nella console Amazon S3. Per ulteriori informazioni, consulta Configurazione della crittografia predefinita.
- Per visualizzare lo stato della crittografia automatica come intestazione di risposta dell'API Amazon S3 aggiuntiva in AWS Command Line Interface (AWS CLI) e negli AWS SDK, controlla l'intestazione di risposta `x-amz-server-side-encryption` quando utilizzi API di azione a oggetti, come e. PutObjectGetObject

Cosa devo fare per trarre vantaggio da questa modifica?

Non è necessario apportare modifiche alle applicazioni esistenti. Poiché la crittografia predefinita è abilitata per tutti i bucket, tutti i nuovi oggetti caricati in Amazon S3 vengono crittografati automaticamente.

Posso disabilitare la crittografia per i nuovi oggetti che vengono scritti nel mio bucket?

No. SSE-S3 è il nuovo livello di crittografia di base che viene applicato a tutti i nuovi oggetti caricati nel bucket. Non è più possibile disabilitare la crittografia per il caricamento di nuovi oggetti.

Ciò avrà ripercussioni sui miei addebiti?

No. La crittografia predefinita con SSE-S3 è disponibile senza costi aggiuntivi. Ti verranno fatturati lo spazio di archiviazione, le richieste e le altre funzionalità di S3, come al solito. Per informazioni sui prezzi, consulta Prezzi di Amazon S3.

Amazon S3 crittograferà i miei oggetti esistenti non crittografati?

No. A partire dal 5 gennaio 2023, Amazon S3 crittografa automaticamente solo i caricamenti di nuovi oggetti. Per crittografare gli oggetti esistenti, è possibile utilizzare la funzionalità Operazioni in batch Amazon S3 per creare copie crittografate degli oggetti. Queste copie crittografate manterranno i dati e il nome dell'oggetto esistenti e verranno crittografate utilizzando le chiavi di crittografia specificate. Per ulteriori informazioni, consulta Encrypting objects with Amazon S3 Batch Operations (Crittografia degli oggetti con Operazioni in batch Amazon S3) in AWS Storage Blog (Blog sull'archiviazione AWS).

Non ho abilitato la crittografia per i miei bucket prima di questa versione. Devo cambiare la modalità di accesso agli oggetti?

No. La crittografia predefinita con SSE-S3 consente di crittografare automaticamente i dati durante la scrittura in Amazon S3 e di eseguire la decrittografia al momento dell'accesso. Non vi è alcuna modifica nel modo in cui si accede agli oggetti crittografati automaticamente.

Devo cambiare il modo in cui accedo ai miei oggetti con crittografia lato client?

No. Tutti gli oggetti con crittografia lato client crittografati prima di essere caricati in Amazon S3 arrivano come oggetti di testo criptato crittografati all'interno di Amazon S3. Questi oggetti avranno ora un livello di crittografia SSE-S3 aggiuntivo. I carichi di lavoro che utilizzano oggetti con crittografia lato client non richiederanno alcuna modifica ai servizi client o alle impostazioni di autorizzazione.

> **Note**
>
> HashiCorp Gli utenti Terraform che non utilizzano una versione aggiornata del AWS Provider potrebbero riscontrare una variazione inaspettata dopo la creazione di nuovi bucket S3 senza una configurazione di crittografia definita dal cliente. Per evitare questa deriva, aggiorna la tua versione di Terraform AWS Provider a una delle seguenti versioni: qualsiasi versione, o. 4.x 3.76.1 2.70.4

Uso della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3)

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva negli SDK and. AWS Command Line Interface AWS Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

Tutti i nuovi caricamenti di oggetti su bucket Amazon S3 vengono crittografati per impostazione predefinita con la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).

La crittografia lato server protegge i dati inattivi. Amazon S3 crittografa ogni oggetto con una chiave univoca. Come ulteriore tutela, crittografa la chiave con una chiave che ruota con regolarità. La crittografia lato server di Amazon S3 utilizza la modalità contatore Advanced Encryption Standard Galois (AES-GCM) a 256 bit per crittografare tutti gli oggetti caricati.

Non sono previsti costi aggiuntivi per l'utilizzo della crittografia lato server con le chiavi gestite da Amazon S3 (SSE-S3). Tuttavia, per le richieste di configurare la funzione di crittografia predefinita vengono applicati i costi delle richieste Amazon S3 standard. Per informazioni sui prezzi, consulta Prezzi di Amazon S3.

Se desideri che i tuoi caricamenti di dati siano crittografati utilizzando solo le chiavi gestite da Amazon S3, puoi utilizzare la seguente policy dei bucket. Ad esempio, la seguente policy del bucket rifiuta le autorizzazioni al caricamento di un oggetto a meno che la richiesta non includa l'intestazione `x-amz-server-side-encryption` per richiedere la codifica lato server:

```
{
  "Version": "2012-10-17",
  "Id": "PutObjectPolicy",
  "Statement": [
    {
      "Sid": "DenyObjectsThatAreNotSSES3",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::example-s3-bucket/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "AES256"
        }
      }
    }
   ]
}
```

> **Note**
>
> La crittografia lato server viene applicata solo ai dati dell'oggetto, non dei metadati dell'oggetto.

Supporto API per la crittografia lato server

Tutti i bucket Amazon S3 hanno la crittografia configurata per impostazione predefinita e tutti i nuovi oggetti caricati in un bucket S3 vengono automaticamente crittografati quando sono a riposo. La crittografia lato server con le chiavi gestite da Amazon S3 (SSE-S3) è la configurazione predefinita della crittografia per ogni bucket di Amazon S3. Per utilizzare un diverso tipo di crittografia, puoi specificare il tipo di crittografia lato server da utilizzare nelle richieste PUT S3 oppure impostare la configurazione di crittografia predefinita nel bucket di destinazione.

Se desideri specificare un tipo di crittografia diverso nelle tue PUT richieste, puoi utilizzare la crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS), la

crittografia lato server a due livelli con AWS KMS chiavi (DSSE-KMS) o la crittografia lato server con chiavi fornite dal cliente (SSE-C). Per impostare una configurazione di crittografia predefinita diversa nel bucket di destinazione puoi utilizzare SSE-KMS o DSSE-KMS.

Per configurare la crittografia lato server utilizzando le REST API della creazione dell'oggetto, devi fornire l'intestazione della richiesta `x-amz-server-side-encryption`. Per ulteriori informazioni sulle APIs REST, consulta Utilizzo di REST API.

Le seguenti API Amazon S3 supportano questa intestazione:

- Operazioni PUT: specifica l'intestazione della richiesta quando si caricano i dati utilizzando l'API PUT. Per ulteriori informazioni, consulta PUT Object.
- Avvia caricamento in più parti: specifica l'intestazione nella richiesta di avvio quando si caricano oggetti di grandi dimensioni utilizzando l'API per il caricamento in più parti. Per ulteriori informazioni, consulta Initiate Multipart Upload.
- Operazione COPY: l'operazione di copia di un oggetto coinvolge un oggetto di origine e un oggetto di destinazione. Per ulteriori informazioni, consulta PUT Object - Copy.

> **Note**
>
> Quando si utilizza un'operazione `POST` per caricare un oggetto anziché l'intestazione della richiesta, si specificano le stesse informazioni nei campi del modulo. Per ulteriori informazioni, consulta POST Object.

Gli SDK forniscono anche API wrapper che puoi utilizzare per richiedere la crittografia lato server. AWS Puoi anche utilizzare il per caricare oggetti e richiedere la crittografia AWS Management Console lato server.

Per ulteriori informazioni generali, consulta Concetti di AWS KMS nella Guida per gli sviluppatori di AWS Key Management Service .

Argomenti

- Specifica della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3)

Specifica della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3)

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva negli SDK and. AWS Command Line Interface AWS Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

Tutti i bucket Amazon S3 hanno la crittografia configurata per impostazione predefinita e tutti i nuovi oggetti caricati in un bucket S3 vengono automaticamente crittografati quando sono a riposo. La crittografia lato server con le chiavi gestite da Amazon S3 (SSE-S3) è la configurazione predefinita della crittografia per ogni bucket di Amazon S3. Per utilizzare un diverso tipo di crittografia, puoi specificare il tipo di crittografia lato server da utilizzare nelle richieste PUT S3 oppure impostare la configurazione di crittografia predefinita nel bucket di destinazione.

Se desideri specificare un tipo di crittografia diverso nelle tue PUT richieste, puoi utilizzare la crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS), la crittografia lato server a due livelli con AWS KMS chiavi (DSSE-KMS) o la crittografia lato server con chiavi fornite dal cliente (SSE-C). Per impostare una configurazione di crittografia predefinita diversa nel bucket di destinazione puoi utilizzare SSE-KMS o DSSE-KMS.

Puoi specificare SSE-S3 utilizzando la console S3, le API REST, gli SDK e (). AWS AWS Command Line Interface AWS CLI Per ulteriori informazioni, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3.

Utilizzo della console S3

Questo argomento descrive in che modo impostare o modificare il tipo di crittografia che viene utilizzato da un oggetto utilizzando la AWS Management Console. Quando si copia un oggetto utilizzando la console, l'oggetto viene copiato da Amazon S3 così com'è: Ciò significa che se l'oggetto di origine è crittografato, anche l'oggetto di destinazione sarà crittografato. Puoi usare la console per aggiungere o modificare la crittografia per un oggetto.

> **Note**
>
> - Se si modifica la crittografia di un oggetto, viene creato un nuovo oggetto per sostituire quello precedente. Se è abilitata la funzione Controllo delle versioni S3, viene creata una nuova versione dell'oggetto e l'oggetto esistente diventa una versione precedente. Il ruolo che modifica la proprietà diventa anche il proprietario del nuovo oggetto o della versione dell'oggetto.
> - Se modificate il tipo di crittografia per un oggetto con tag definiti dall'utente, dovete disporre dell'autorizzazione. `s3:GetObjectTagging` Se state modificando il tipo di crittografia per un oggetto che non ha tag definiti dall'utente ma ha una dimensione superiore a 16 MB, è necessario disporre anche dell'`s3:GetObjectTagging`autorizzazione.
>
> Se la policy del bucket di destinazione nega l'`s3:GetObjectTagging`azione, il tipo di crittografia dell'oggetto verrà aggiornato, ma i tag definiti dall'utente verranno rimossi dall'oggetto e verrà visualizzato un errore.

Per modificare la crittografia di un oggetto

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckets (Bucket) scegliere il nome del bucket contenente l'oggetto.
4. Nell'elenco Nome scegli il nome dell'oggetto per cui desideri aggiungere o modificare la crittografia.

   Viene visualizzata la pagina dei dettagli dell'oggetto, con diverse sezioni che visualizzano le proprietà dell'oggetto.
5. Scegliere la scheda Properties (Proprietà).
6. Scorri verso il basso fino alla sezione Impostazioni crittografia lato server, quindi scegli Modifica.
7. In Impostazioni di crittografia scegli Usa impostazioni di crittografia predefinite del bucket o Sostituisci impostazioni di crittografia predefinite del bucket.
8. Se scegli Sostituisci impostazioni del bucket per la crittografia predefinita, configura le seguenti impostazioni di crittografia.

- Per Tipo di crittografia scegli Chiavi gestite da Amazon S3 (SSE-S3). Per crittografare gli oggetti, SSE-S3 utilizza una delle cifrature di blocco più complesse, lo standard di crittografia avanzata a 256 bit (AES-256). Per ulteriori informazioni, consulta Uso della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).

9. Seleziona Save changes (Salva modifiche).

> **Note**
>
> Questa azione applica la crittografia a tutti gli oggetti specificati. Durante la crittografia delle cartelle, attendere il completamento dell'operazione di salvataggio prima di aggiungere nuovi oggetti alla cartella.

Utilizzo di REST API

Al momento della creazione dell'oggetto, ovvero quando si carica un nuovo oggetto o si esegue una copia di un oggetto esistente, è possibile specificare se si desidera che Amazon S3 esegua la crittografia dei dati con le chiavi gestite da Amazon S3 (SSE-S3) aggiungendo alla richiesta l'intestazione `x-amz-server-side-encryption`. Imposta il valore dell'intestazione sull'algoritmo della crittografia `AES256` supportato da Amazon S3. Amazon S3 conferma che l'oggetto è stato archiviato utilizzando SSE-S3 restituendo l'intestazione della risposta `x-amz-server-side-encryption`.

Le operazioni API per il caricamento REST elencate di seguito accettano l'intestazione della richiesta `x-amz-server-side-encryption`.

- PUT Object
- PUT Object - Copy
- POST Object
- Avvio del caricamento in più parti

Quando si caricano oggetti di grandi dimensioni utilizzando l'operazione API per il caricamento in più parti, è possibile specificare la crittografia lato server aggiungendo l'intestazione `x-amz-server-side-encryption` alla richiesta di avvio del caricamento in più parti. Quando si copia un oggetto esistente, indipendentemente dal fatto che l'oggetto di origine sia stato o meno crittografato, l'oggetto

di destinazione non viene crittografato, a meno che non si richieda esplicitamente la crittografia lato server.

Quando un oggetto viene archiviato utilizzando SSE-S3, le intestazioni di risposta delle seguenti operazioni REST API restituiscono l'intestazione `x-amz-server-side-encryption`.

- PUT Object
- PUT Object - Copy
- POST Object
- Avvio del caricamento in più parti
- Upload Part
- Caricamento di parte - Copy
- Completamento del caricamento in più parti
- Get Object
- Head Object

> **Note**
>
> Non inviare l'intestazione di richiesta di crittografia per richieste `GET` e `HEAD` se l'oggetto utilizza SSE-S3 per evitare di ricevere un errore HTTP 400 (Bad Request).

Utilizzo degli SDK AWS

Quando usi AWS gli SDK, puoi richiedere ad Amazon S3 di utilizzare la crittografia lato server con le chiavi di crittografia gestite di Amazon S3 (SSE-S3). Questa sezione fornisce esempi di utilizzo degli SDK in più lingue. AWS Per ulteriori informazioni su altri SDK, consulta Codici di esempio e librerie.

Java

Quando si utilizza il AWS SDK for Java per caricare un oggetto, è possibile utilizzare SSE-S3 per crittografarlo. Per richiedere la crittografia lato server, utilizza la proprietà `ObjectMetadata` della `PutObjectRequest` per impostare l'intestazione della richiesta `x-amz-server-side-encryption`. Quando si utilizza il metodo `putObject()` di `AmazonS3Client`, Amazon S3 cripta i dati e li salva.

È anche possibile richiedere la crittografia SSE-S3 durante il caricamento di oggetti con l'operazione API per il caricamento in più parti:

- Quando si utilizza l'operazione API per il caricamento in più parti di alto livello, usi i metodi `TransferManager` per applicare la crittografia lato server agli oggetti durante il loro caricamento. È possibile utilizzare uno qualsiasi dei metodi di caricamento che accetta `ObjectMetadata` come parametro. Per ulteriori informazioni, consulta Caricamento di un oggetto utilizzando il caricamento in più parti.
- Quando si utilizza l'operazione API per il caricamento in più parti di basso livello, specifichi la crittografia lato server quando avvii il caricamento in più parti. Si aggiungi la proprietà `ObjectMetadata` mediante una chiamata al metodo `InitiateMultipartUploadRequest.setObjectMetadata()`. Per ulteriori informazioni, consulta Utilizzo degli AWS SDK (API di basso livello).

Non puoi direttamente modificare lo stato di crittografia di un oggetto (la crittografia di un oggetto non crittografato o la decrittografia dell'oggetto crittografato). Per modificare lo stato di crittografia di un oggetto, effettuare una copia dell'oggetto, specificando lo stato di crittografia per la copia e poi eliminare l'oggetto originale. Amazon S3 esegue la crittografia dell'oggetto copiato solo se hai effettuato una richiesta specifica di crittografia lato server. Per richiedere la crittografia dell'oggetto copiato tramite l'API Java, utilizza la proprietà `ObjectMetadata` per specificare la crittografia lato server in `CopyObjectRequest`, come mostrato nell'esempio di codice Java riportato di seguito.

Example Esempio

L'esempio che segue mostra come impostare la crittografia lato server utilizzando AWS SDK for Java. Mostra come eseguire le seguenti operazioni:

- Carica un nuovo oggetto usando SSE-S3.
- Modifica lo stato di crittografia di un oggetto (in questo esempio, crittografare un oggetto precedentemente non crittografato) eseguendo una copia dell'oggetto.
- Controlla lo stato di crittografia dell'oggetto.

Per ulteriori informazioni sulla crittografia lato server, consulta Utilizzo di REST API. Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella Developer Guide. 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

Quando carichi un oggetto, puoi indirizzare Amazon S3 per crittografarlo. Per modificare lo stato di crittografia di un oggetto esistente, effettuare una copia dell'oggetto ed eliminare l'oggetto di origine. Per impostazione predefinita, l'operazione di copia crittografa la destinazione solo se si richiede esplicitamente la crittografia lato server dell'oggetto di destinazione. Per specificare SSE-S3 in `CopyObjectRequest`, aggiungi quanto segue:

```
ServerSideEncryptionMethod = ServerSideEncryptionMethod.AES256
```

Per un esempio di come copiare un oggetto, consulta Utilizzo degli AWS SDK.

Nel seguente esempio viene caricato un oggetto. Nella richiesta l'esempio indirizza Amazon S3 per crittografare l'oggetto. L'esempio poi recupera i metadati dell'oggetto e verifica i metodi di crittografia utilizzati. Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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

Questo argomento mostra come utilizzare le classi della versione 3 di AWS SDK for PHP per aggiungere SSE-S3 agli oggetti caricati su Amazon S3. Per ulteriori informazioni sull'API AWS SDK for Ruby, AWS vai a SDK for Ruby - Versione 2.

Per caricare un oggetto su Amazon S3, si utilizza il metodo Aws\S3\S3Client::putObject(). Per aggiungere l'intestazione di richiesta `x-amz-server-side-encryption` alla richiesta di caricamento, specificare il parametro`ServerSideEncryption` con il valore `AES256`, come mostrato nel seguente esempio di codice. Per ulteriori informazioni sulla crittografia lato server delle richieste, consultare Utilizzo di REST API.

```
 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',
]);
```

Nella risposta Amazon S3 restituisce l'intestazione `x-amz-server-side-encryption` con il valore dell'algoritmo di crittografia utilizzato per crittografare i dati dell'oggetto.

Quando si caricano oggetti di grandi dimensioni utilizzando l'API per il caricamento in più parti, puoi specificare SSE-S3 per gli oggetti che carichi, come segue:

- Quando utilizzi l'operazione API di caricamento multiparte di basso livello, specifica la crittografia lato server quando chiami il metodo Aws\ S3\ S3Client:: (). createMultipartUpload Per aggiungere l'intestazione di richiesta `x-amz-server-side-encryption` alla richiesta, specificare la chiave `array` del parametro `ServerSideEncryption` con il valore `AES256`. Per ulteriori informazioni sull'operazione API per il caricamento in più parti di basso livello, consulta Utilizzo degli AWS SDK (API di basso livello).
- Quando utilizzi l'operazione API di caricamento multiparte di alto livello, specifica la crittografia lato server utilizzando il parametro dell'operazione API. `ServerSideEncryption` CreateMultipartUpload Per un esempio di utilizzo del metodo `setOption()` con l'operazione API per il caricamento in più parti di alto livello, consulta Caricamento di un oggetto utilizzando il caricamento in più parti.

Per determinare lo stato di crittografia di un oggetto esistente, recuperare i metadati dell'oggetto richiamando il metodo Aws\S3\S3Client::headObject(), come mostrato nell'esempio di codice PHP riportato di seguito.

```
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'];
```

Per modificare lo stato di crittografia di un oggetto esistente, effettuare una copia dell'oggetto utilizzando il metodo Aws\S3\S3Client::copyObject() ed eliminare l'oggetto di origine. Per impostazione predefinita, `copyObject()` non esegue la crittografia della destinazione, a meno che non si richieda esplicitamente la crittografia lato server dell'oggetto di destinazione utilizzando il parametro `ServerSideEncryption` con il valore `AES256`. Il seguente esempio di codice PHP esegue una copia di un oggetto e aggiunge la crittografia lato server all'oggetto copiato.

```
 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',
]);
```

Per ulteriori informazioni, consulta i seguenti argomenti:

- AWS SDK for PHP per la classe Amazon S3 Aws\ S3\ S3Client
- Documentazione AWS SDK for PHP

Ruby

Quando si utilizza AWS SDK for Ruby per caricare un oggetto, è possibile specificare che l'oggetto venga archiviato crittografato quando è inattivo con SSE-S3. Dopo essere stato letto, l'oggetto viene automaticamente decrittografato.

Il seguente esempio di AWS SDK for Ruby versione 3 dimostra come specificare che un file caricato su Amazon S3 sia crittografato quando è inattivo.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectPutSseWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object_encrypted(object_content, encryption)
    @object.put(body: object_content, server_side_encryption: encryption)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put your content to #{object.key}. Here's why: #{e.message}"
    false
  end
end

# Example usage:
```

```
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-encrypted-content"
  object_content = "This is my super-secret content."
  encryption = "AES256"

  wrapper = ObjectPutSseWrapper.new(Aws::S3::Object.new(bucket_name,
 object_content))
  return unless wrapper.put_object_encrypted(object_content, encryption)

  puts "Put your content into #{bucket_name}:#{object_key} and encrypted it with
 #{encryption}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

L'esempio di codice seguente dimostra come determinare lo stato di crittografia di un oggetto esistente.

```
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__
```

Se la crittografia lato server non viene utilizzata per l'oggetto archiviato in Amazon S3, il metodo restituisce `null`.

Per modificare lo stato di crittografia di un oggetto esistente, effettuare una copia dell'oggetto ed eliminare l'oggetto di origine. Per default, i metodi di copia non eseguono la crittografia della destinazione, a meno che non si richieda esplicitamente la crittografia lato server. È possibile richiedere la crittografia dell'oggetto di destinazione specificando il valore `server_side_encryption` nell'argomento hash dell'opzione, come mostrato nel seguente esempio di codice Ruby. L'esempio di codice dimostra come copiare un oggetto e crittografare 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__
```

Usando il AWS CLI

Per specificare SSE-S3 quando caricate un oggetto utilizzando il AWS CLI, utilizzate il seguente esempio.

```
aws s3api put-object --bucket example-s3-bucket1 --key object-key-name --server-side-
encryption AES256  --body file path
```

Per ulteriori informazioni, consulta put-object in Riferimenti della AWS CLI . Per specificare SSE-S3 quando si copia un oggetto utilizzando il, vedere copy-object. AWS CLI

Usando AWS CloudFormation

Per esempi di configurazione della crittografia utilizzando AWS CloudFormation, consulta l'esempio Creare un bucket con crittografia predefinita e Creare un bucket utilizzando la crittografia AWS KMS lato server con una chiave S3 Bucket nell'argomento della Guida per l'*`AWS::S3::Bucket ServerSideEncryptionRule`*utente.AWS CloudFormation

Utilizzo della crittografia lato server con chiavi (SSE-KMS) AWS KMS

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva negli SDK and. AWS Command Line Interface AWS Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

La crittografia lato server è la crittografia dei dati nella posizione di destinazione eseguita dall'applicazione o dal servizio che li riceve.

Amazon S3 abilita automaticamente la crittografia lato server con le chiavi gestite da Amazon S3 (SSE-S3) per il caricamento di nuovi oggetti.

Salvo diversa indicazione, per crittografare gli oggetti i bucket utilizzano SSE-S3 per impostazione predefinita. Tuttavia, puoi scegliere di configurare i bucket per utilizzare invece la crittografia lato server con () chiavi (SSE-KMS). AWS Key Management Service AWS KMS Per ulteriori informazioni, consulta Specifica della crittografia lato server con AWS KMS (SSE-KMS).

AWS KMS è un servizio che combina hardware e software sicuri e ad alta disponibilità per fornire un sistema di gestione delle chiavi scalabile per il cloud. Amazon S3 utilizza la crittografia lato server con AWS KMS (SSE-KMS) per crittografare i dati degli oggetti S3. Inoltre, quando viene richiesto SSE-KMS per l'oggetto, il checksum S3 (come parte dei metadati dell'oggetto) viene archiviato in forma crittografata. Per ulteriori informazioni sui checksum, consulta Verifica dell'integrità degli oggetti.

Se utilizzi le chiavi KMS, puoi utilizzarle AWS KMS tramite l'AWS Management Consoleo l'API per effettuare le seguenti operazioni:AWS KMS

- Creare, visualizzare, modificare, monitorare, abilitare o disabilitare, ruotare e pianificare l'eliminazione delle chiavi KMS in modo centralizzato.
- Definire le policy che controllano come e da chi possono essere utilizzate le chiavi KMS.
- Verificare il loro utilizzo per dimostrare che sono state utilizzate correttamente. Il controllo è supportato dall'API AWS KMS, ma non dalla AWS Management ConsoleAWS KMS.

I controlli di sicurezza inclusi AWS KMS possono aiutarti a soddisfare i requisiti di conformità relativi alla crittografia. Puoi utilizzare queste chiavi KMS per proteggere i dati nei bucket Amazon S3. Quando utilizzi la crittografia SSE-KMS con un bucket S3, AWS KMS keys deve trovarsi nella stessa regione del bucket.

Sono previsti costi aggiuntivi per l'utilizzo. AWS KMS keys Per ulteriori informazioni, consulta la sezione Concetti di AWS KMS key nella Guida per gli sviluppatori di AWS Key Management Service e i Prezzi di AWS KMS.

Autorizzazioni

Per caricare un oggetto crittografato con un AWS KMS key su Amazon S3, sono necessarie `kms:GenerateDataKey` le autorizzazioni sulla chiave. Per scaricare un oggetto crittografato con un AWS KMS key, sono necessarie `kms:Decrypt` le autorizzazioni. Per informazioni sulle AWS KMS autorizzazioni necessarie per i caricamenti in più parti, consulta. Autorizzazioni e API per il caricamento in più parti

> **Important**
>
> Esamina attentamente le autorizzazioni concesse nelle politiche chiave del tuo KMS. Limita sempre le autorizzazioni relative alle policy chiave KMS gestite dal cliente solo ai responsabili

e ai AWS servizi IAM che devono accedere all'azione chiave pertinente. AWS KMS Per ulteriori informazioni, consulta Politiche chiave in. AWS KMS

Argomenti

- AWS KMS keys
- Chiavi bucket Amazon S3
- Richiesta della crittografia lato server
- Contesto di crittografia
- Invio di richieste per oggetti AWS KMS crittografati
- Specifica della crittografia lato server con AWS KMS (SSE-KMS)
- Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3

AWS KMS keys

Quando utilizzi la crittografia lato server con AWS KMS (SSE-KMS), puoi utilizzare la chiave AWS gestita predefinita oppure puoi specificare una chiave gestita dal cliente che hai già creato. AWS KMS supporta la crittografia delle buste. S3 utilizza le AWS KMS funzionalità di crittografia delle buste per proteggere ulteriormente i dati. La crittografia a busta consiste nel crittografare i dati di testo non crittografato con una chiave di dati e quindi nel crittografare la chiave di dati con una chiave KMS. Per ulteriori informazioni sulla crittografia envelope, consulta Crittografia envelope nella Guida per sviluppatori di AWS Key Management Service .

Se non specifichi una chiave gestita dal cliente, Amazon S3 ne crea automaticamente una per Account AWS la prima volta che aggiungi un Chiave gestita da AWS oggetto crittografato con SSE-KMS a un bucket. Per impostazione predefinita, Amazon S3 utilizza questa chiave KMS per SSE-KMS.

> **Note**
>
> Gli oggetti crittografati mediante SSE-KMS con Chiavi gestite da AWS non possono essere condivisi tra più account. Se devi condividere i dati SSE-KMS tra più account, devi utilizzare una chiave gestita dal cliente da. AWS KMS

Se desideri utilizzare una chiave gestita dal cliente per SSE-KMS, crea una chiave di crittografia simmetrica gestita dal cliente prima di configurare SSE-KMS. Quindi, quando configuri SSE-KMS per il bucket, potrai specificare la chiave gestita dal cliente esistente. Per ulteriori informazioni sulla chiave di crittografia simmetrica, consulta Chiavi KMS di crittografia simmetrica nella Guida per gli sviluppatori di AWS Key Management Service .

La creazione di una chiave gestita dal cliente offre maggiore flessibilità e controllo. Ad esempio, puoi creare, ruotare e disabilitare le chiavi gestite dal cliente. Puoi anche definire controlli di accesso e controllare le chiavi gestite dal cliente utilizzate per proteggere i dati. Per ulteriori informazioni sulle chiavi gestite e AWS gestite dal cliente, consulta Customer keys and AWS keys nella Developer Guide.AWS Key Management Service

> **Note**
>
> Quando utilizzi la crittografia lato server con una chiave gestita dal cliente archiviata in un archivio di chiavi esterno, a differenza delle chiavi KMS standard, hai la responsabilità di garantire la disponibilità e la durata del materiale chiave. Per ulteriori informazioni sugli archivi di chiavi esterni e sul loro impatto sul modello di responsabilità condivisa, vedi Archivi di chiavi esterni nella Guida per gli sviluppatori di AWS Key Management Service .

Utilizzo della crittografia SSE-KMS per operazioni multi-account

Quando si utilizza la crittografia per operazioni multi-account, tieni presente quanto segue:

- Se non viene fornito un AWS KMS key Amazon Resource Name (ARN) o un alias al momento della richiesta o tramite la configurazione di crittografia predefinita del bucket, viene utilizzata la Chiave gestita da AWS ()`aws/s3`.
- Se stai caricando o accedendo a oggetti S3 utilizzando principi AWS Identity and Access Management (IAM) che sono gli stessi Account AWS della tua chiave KMS, puoi usare il (). Chiave gestita da AWS `aws/s3`
- Se desideri concedere l'accesso multi-account agli oggetti S3, utilizza una chiave gestita dal cliente. Puoi configurare la policy di una chiave gestita dal cliente per consentire l'accesso da un altro account.
- Se stai specificando una chiave KMS gestita dal cliente, ti consigliamo di utilizzare una chiave KMS ARN completamente qualificata. Se invece utilizzi un alias di chiave KMS, AWS KMS risolve la chiave all'interno dell'account del richiedente. Ciò potrebbe comportare la crittografia dei dati con una chiave KMS di proprietà del richiedente e non del proprietario del bucket.

- È necessario specificare una chiave per cui il richiedente ha ottenuto l'autorizzazione `Encrypt`. Per ulteriori informazioni, consulta l'argomento relativo all'autorizzazione concessa agli utenti delle chiavi di utilizzare una chiave KMS per le operazioni di crittografia nella Guida per gli sviluppatori di AWS Key Management Service .

Per ulteriori informazioni su quando utilizzare le chiavi gestite dal cliente e le chiavi KMS AWS gestite, consulta Devo usare una chiave Chiave gestita da AWS o una chiave gestita dal cliente per crittografare i miei oggetti in Amazon S3?

Flusso di lavoro di crittografia SSE-KMS

Se scegli di crittografare i tuoi dati utilizzando una chiave Chiave gestita da AWS o una chiave gestita dal cliente AWS KMS e Amazon S3 esegue le seguenti azioni di crittografia della busta:

1. Amazon S3 richiede una chiave di dati in testo non formattato e una copia della chiave crittografata con la chiave KMS specificata.
2. AWS KMS genera una chiave dati, la crittografa con la chiave KMS e invia sia la chiave dati in testo semplice che la chiave dati crittografata ad Amazon S3.
3. Amazon S3 crittografa i dati utilizzando la chiave di dati ed eliminando appena possibile la chiave di testo normale dalla memoria dopo l'utilizzo.
4. Amazon S3 archivia la chiave di dati crittografata come metadati con i dati crittografati.

Quando richiedi che i tuoi dati vengano decrittografati, usa Amazon S3 AWS KMS ed esegui le seguenti azioni:

1. Amazon S3 invia la chiave dati crittografata AWS KMS a una `Decrypt` richiesta.
2. AWS KMS decrittografa la chiave dati crittografata utilizzando la stessa chiave KMS e restituisce la chiave dati in testo semplice ad Amazon S3.
3. Amazon S3 utilizza la chiave di dati non crittografati per decrittografare i dati crittografati, quindi rimuove il prima possibile la chiave di dati non crittografati dalla memoria.

> **Important**
>
> Quando utilizzi una chiave KMS AWS KMS key per la crittografia lato server in Amazon S3, devi scegliere una chiave KMS di crittografia simmetrica. Amazon S3 supporta solo chiavi

KMS di crittografia simmetrica. Per ulteriori informazioni sulle chiavi, consulta Chiavi KMS di crittografia simmetrica nella Guida per gli sviluppatori di AWS Key Management Service .

Verifica della crittografia SSE-KMS

Per identificare le richieste che specificano SSE-KMS, puoi utilizzare i parametri All SSE-KMS requests (Tutte le richieste SSE-KMS) e % all SSE-KMS requests (% tutte le richieste SSE-KMS) nei parametri di Amazon S3 Storage Lens. S3 Storage Lens è una funzionalità di analisi dell'archiviazione su cloud che puoi utilizzare per avere una panoramica completa a livello di organizzazione sull'utilizzo e sulle attività relative all'archiviazione di oggetti. Puoi anche utilizzare il numero di bucket abilitati per SSE-KMS e i bucket abilitati per% SSE-KMS per comprendere il numero di bucket abilitati (SSE-KMS) per la crittografia dei bucket predefinita. Per ulteriori informazioni, consulta Valutazione dell'attività e dell'utilizzo dello storage con S3 Storage Lens. Per un elenco completo dei parametri, consulta Glossario dei parametri di S3 Storage.

Per verificare l'utilizzo delle chiavi per i dati crittografati SSE-KMS, è possibile utilizzare i log. AWS KMS AWS CloudTrail Puoi ottenere informazioni dettagliate sulle tue operazioni crittografiche, ad esempio e. GenerateDataKeyDecrypt CloudTrail supporta numerosi valori di attributo per filtrare la ricerca, tra cui il nome dell'evento, il nome utente e l'origine dell'evento.

Chiavi bucket Amazon S3

Quando configuri la crittografia lato server utilizzando AWS KMS (SSE-KMS), puoi configurare i bucket per utilizzare S3 Bucket Keys per SSE-KMS. L'utilizzo di una chiave a livello di bucket per SSE-KMS può ridurre i costi delle AWS KMS richieste fino al 99 percento diminuendo il traffico delle richieste da Amazon S3 a. AWS KMS

Quando si configura il bucket per utilizzare una chiave di bucket S3 per SSE-KMS su nuovi oggetti, AWS KMS genera una chiave a livello di bucket che viene utilizzata per creare chiavi di dati univoche per gli oggetti nel bucket. Questa S3 Bucket Key viene utilizzata per un periodo di tempo limitato all'interno di Amazon S3, riducendo ulteriormente la necessità per Amazon S3 di effettuare richieste per completare le operazioni di crittografia. AWS KMS Per ulteriori informazioni sull'utilizzo delle chiavi del bucket S3, consulta la sezione Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3.

Richiesta della crittografia lato server

Per richiedere la crittografia lato server di tutti gli oggetti in uno specifico bucket Amazon S3, è possibile utilizzare una policy di bucket. Ad esempio, la seguente policy di bucket rifiuta a

chiunque l'autorizzazione al caricamento dell'oggetto (`s3:PutObject`) se la richiesta non include un'intestazione `x-amz-server-side-encryption-aws-kms-key-id` che richiede la crittografia lato server con SSE-KMS.

```
{
   "Version":"2012-10-17",
   "Id":"PutObjectPolicy",
   "Statement":[{
         "Sid":"DenyObjectsThatAreNotSSEKMS",
         "Effect":"Deny",
         "Principal":"*",
         "Action":"s3:PutObject",
         "Resource":"arn:aws:s3:::example-s3-bucket1/*",
         "Condition":{
            "Null":{
               "s3:x-amz-server-side-encryption-aws-kms-key-id":"true"
            }
         }
      }
   ]
}
```

Per richiedere che un particolare AWS KMS key venga utilizzato per crittografare gli oggetti in un bucket, puoi usare la chiave condition. `s3:x-amz-server-side-encryption-aws-kms-key-id` Per specificare la chiave KMS, devi utilizzare una chiave Amazon Resource Name (ARN) nel `arn:aws:kms:region:acct-id:key/key-id` formato. AWS Identity and Access Management non convalida se la stringa for esiste. `s3:x-amz-server-side-encryption-aws-kms-key-id`

> **Note**
>
> Quando carichi un oggetto, puoi specificare la chiave KMS con l'intestazione `x-amz-server-side-encryption-aws-kms-key-id`. Se l'intestazione non è presente nella richiesta, Amazon S3 presuppone che voglia utilizzare la Chiave gestita da AWS. Indipendentemente da ciò, l'ID della AWS KMS chiave utilizzato da Amazon S3 per la crittografia degli oggetti deve corrispondere all'ID della AWS KMS chiave nella policy, altrimenti Amazon S3 nega la richiesta.

Per un elenco completo delle chiavi di condizione specifiche di Amazon S3, consulta Condition keys for Amazon S3 nel Service Authorization Reference.

Contesto di crittografia

Un contesto di crittografia è un set di coppie chiave-valore che contiene ulteriori informazioni contestuali sui dati. Il contesto di crittografia non è crittografato. Quando viene specificato un contesto di crittografia per un'operazione di crittografia, Amazon S3 deve specificare lo stesso contesto di crittografia per l'operazione di decrittografia. In caso contrario, la decrittografia non riesce. AWS KMS utilizza il contesto di crittografia come dati autenticati aggiuntivi (AAD) per supportare la crittografia autenticata. Per ulteriori informazioni sul contesto di crittografia, consulta il Contesto di crittografia nella Guida per gli sviluppatori di AWS Key Management Service .

Per impostazione predefinita, Amazon S3 utilizza il nome della risorsa Amazon (ARN) dell'oggetto o del bucket come coppia di contesto di crittografia:

- Se utilizzi SSE-KMS senza abilitare una chiave bucket S3, l'ARN del oggetto viene utilizzato come contesto di crittografia.

  ```
  arn:aws:s3:::object_ARN
  ```

- Se utilizzi SSE-KMS e abiliti una chiave di bucket S3, l'ARN del bucket viene utilizzato come contesto di crittografia. Per ulteriori informazioni sui bucket S3, consulta la sezione Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3.

  ```
  arn:aws:s3:::bucket_ARN
  ```

Facoltativamente, puoi fornire una coppia di contesti di crittografia aggiuntiva utilizzando l'`x-amz-server-side-encryption-context`intestazione in una richiesta s3:. PutObject Tuttavia, poiché il contesto di crittografia non è crittografato, assicurati che non includa informazioni sensibili. Amazon S3 archivia questa coppia di chiavi aggiuntiva insieme al contesto di crittografia predefinito. Quando elabora la tua richiesta PUT, Amazon S3 aggiunge il contesto di crittografia predefinito di `aws:s3:arn` a quello che fornisci.

È possibile utilizzare il contesto di crittografia per identificare e categorizzare le operazioni di crittografia. Puoi anche utilizzare il valore ARN del contesto di crittografia predefinito per tenere traccia delle richieste pertinenti AWS CloudTrail visualizzando quale ARN Amazon S3 è stato utilizzato con quale chiave di crittografia.

Nel `requestParameters` campo di un file di CloudTrail registro, il contesto di crittografia è simile al seguente.

```
"encryptionContext": {
    "aws:s3:arn": "arn:aws:s3:::example-s3-bucket1/file_name"
}
```

Quando utilizzi SSE-KMS con la funzione opzionale chiavi bucket S3, il valore di contesto di crittografia è l'ARN del bucket.

```
"encryptionContext": {
    "aws:s3:arn": "arn:aws:s3:::example-s3-bucket1"
}
```

Invio di richieste per oggetti AWS KMS crittografati

> **Important**
>
> Tutte `GET` le `PUT` richieste di oggetti AWS KMS crittografati devono essere effettuate utilizzando Secure Sockets Layer (SSL) o Transport Layer Security (TLS). Le richieste devono inoltre essere firmate utilizzando credenziali valide, come AWS Signature Version 4 (o AWS Signature Version 2).

AWS Signature Version 4 è il processo di aggiunta di informazioni di autenticazione alle AWS richieste inviate tramite HTTP. Per motivi di sicurezza, la maggior parte delle richieste AWS deve essere firmata con una chiave di accesso, che consiste in un ID della chiave di accesso e una chiave di accesso segreta. Queste due chiavi in genere vengono definite come le tue credenziali di sicurezza. Per ulteriori informazioni, consulta le sezioni Autenticazione delle richieste (AWS Signature Version 4) e Processo di firma Signature Version 4.

> **Important**
>
> Se l'oggetto utilizza SSE-KMS, non inviare intestazioni di richiesta di crittografia per le richieste `GET` e `HEAD`. In caso contrario, riceverai un errore HTTP 400 Bad Request (HTTP 400 - Richiesta non valida).

Argomenti

Specifica della crittografia lato server con AWS KMS (SSE-KMS)

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva negli SDK and. AWS Command Line Interface AWS Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

Tutti i bucket Amazon S3 hanno la crittografia configurata per impostazione predefinita e tutti i nuovi oggetti caricati in un bucket S3 vengono automaticamente crittografati quando sono a riposo. La crittografia lato server con le chiavi gestite da Amazon S3 (SSE-S3) è la configurazione predefinita della crittografia per ogni bucket di Amazon S3. Per utilizzare un diverso tipo di crittografia, puoi specificare il tipo di crittografia lato server da utilizzare nelle richieste PUT S3 oppure impostare la configurazione di crittografia predefinita nel bucket di destinazione.

Se desideri specificare un tipo di crittografia diverso nelle tue PUT richieste, puoi utilizzare la crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS), la crittografia lato server a due livelli con AWS KMS chiavi (DSSE-KMS) o la crittografia lato server con chiavi fornite dal cliente (SSE-C). Per impostare una configurazione di crittografia predefinita diversa nel bucket di destinazione puoi utilizzare SSE-KMS o DSSE-KMS.

È possibile applicare la crittografia quando stai caricando un nuovo oggetto o copiando un oggetto esistente.

Puoi specificare SSE-KMS utilizzando la console Amazon S3, le operazioni API REST, gli AWS SDK e (). AWS Command Line Interface AWS CLI Per ulteriori informazioni, consulta i seguenti argomenti.

> **Note**
>
> Puoi usare più regioni AWS KMS keys in Amazon S3. Tuttavia, Amazon S3 attualmente tratta le chiavi multi-regione come se fossero chiavi a regione singola e non utilizza le caratteristiche multi-regione della chiave. Per ulteriori informazioni, consulta Utilizzo delle chiavi multi-regione nella Guida per gli sviluppatori di AWS Key Management Service .

> **Note**
>
> Se desideri utilizzare una chiave KMS di proprietà di un altro account, devi disporre dell'autorizzazione per utilizzare la chiave. Per ulteriori informazioni sulle autorizzazioni tra account per le chiavi KMS, vedi Creazione di chiavi KMS utilizzabili da altri account nella Guida per gli sviluppatori di AWS Key Management Service .

Utilizzo della console S3

Questo argomento descrive come impostare o modificare il tipo di crittografia di un oggetto per utilizzare la crittografia lato server con AWS Key Management Service (AWS KMS) chiavi (SSE-KMS) utilizzando la console Amazon S3.

> **Note**
>
> - Se si modifica la crittografia di un oggetto, viene creato un nuovo oggetto per sostituire quello precedente. Se è abilitata la funzione Controllo delle versioni S3, viene creata una nuova versione dell'oggetto e l'oggetto esistente diventa una versione precedente. Il ruolo che modifica la proprietà diventa anche il proprietario del nuovo oggetto o della versione dell'oggetto.
> - Se modifichi il tipo di crittografia per un oggetto con tag definiti dall'utente, devi disporre dell'autorizzazione. `s3:GetObjectTagging` Se state modificando il tipo di crittografia per un oggetto che non ha tag definiti dall'utente ma ha una dimensione superiore a 16 MB, è necessario disporre anche dell'`s3:GetObjectTagging`autorizzazione.
>
> Se la policy del bucket di destinazione nega l'`s3:GetObjectTagging`azione, il tipo di crittografia dell'oggetto verrà aggiornato, ma i tag definiti dall'utente verranno rimossi dall'oggetto e verrà visualizzato un errore.

Per aggiungere o modificare la crittografia di un oggetto

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckets (Bucket) scegliere il nome del bucket contenente l'oggetto.
4. Nell'elenco Nome scegli il nome dell'oggetto per cui desideri aggiungere o modificare la crittografia.

   Viene visualizzata la pagina dei dettagli dell'oggetto, con diverse sezioni che visualizzano le proprietà dell'oggetto.
5. Scegliere la scheda Properties (Proprietà).
6. Scorri verso il basso fino alla sezione Impostazioni crittografia lato server e scegli Modifica.

   Viene visualizzata la pagina Modifica crittografia lato server.
7. In Crittografia lato server, per Impostazioni di crittografia, scegli Sostituisci impostazioni di crittografia predefinite del bucket.
8. In Tipo di crittografia, scegli Crittografia lato server con AWS Key Management Service chiavi (SSE-KMS).

   > **Important**
   >
   > Se usi l'opzione SSE-KMS per la configurazione della crittografia predefinita, vengono applicati i limiti di richieste al secondo (RPS) pari a AWS KMS. Per ulteriori informazioni sulle quote AWS KMS e su come richiedere un aumento delle quote, consulta Quote nella Guida per gli sviluppatori di AWS Key Management Service .
9. In Chiave AWS KMS effettua una delle seguenti operazioni per scegliere la chiave KMS:

   - Per scegliere da un elenco di chiavi KMS disponibili, seleziona Scegli tra le chiavi AWS KMS keys, quindi scegli la chiave KMS dall'elenco delle chiavi disponibili.

     In questo elenco vengono visualizzate sia la chiave Chiave gestita da AWS (`aws/s3`) che quella gestita dal cliente. Per ulteriori informazioni sulle chiavi gestite dal cliente, consulta Chiavi gestite dal cliente e chiavi AWS nella Guida per gli sviluppatori di AWS Key Management Service .

- Per inserire l'ARN della chiave KMS, scegli Inserisci AWS KMS key ARN, quindi inserisci l'ARN della chiave KMS nel campo visualizzato.
- Per creare una nuova chiave gestita dal cliente nella AWS KMS console, scegli Crea una chiave KMS.

  Per ulteriori informazioni sulla creazione di una AWS KMS key, consulta Creating keys nella AWS Key Management Service Developer Guide.

> **Important**
>
> Puoi utilizzare solo le chiavi KMS disponibili nello Regione AWS stesso bucket. La console Amazon S3 elenca solo le prime 100 chiavi KMS nella stessa regione del bucket. Per utilizzare una chiave KMS non elencata, devi inserire l'ARN della chiave KMS. Se desideri utilizzare una chiave KMS di proprietà di un account diverso, è necessario innanzitutto disporre dell'autorizzazione necessaria per l'uso della chiave e quindi inserire l'ARN della chiave KMS.
> Amazon S3 supporta solo chiavi KMS di crittografia simmetriche e non chiavi KMS asimmetriche. Per ulteriori informazioni, consulta Identificazione delle chiavi KMS simmetriche e asimmetriche nella Guida per gli sviluppatori di AWS Key Management Service .

10. Seleziona Salva modifiche.

> **Note**
>
> Questa azione applica la crittografia a tutti gli oggetti specificati. Durante la crittografia delle cartelle, attendere il completamento dell'operazione di salvataggio prima di aggiungere nuovi oggetti alla cartella.

Utilizzo di REST API

Quando crei un oggetto, ovvero quando carichi un nuovo oggetto o copi un oggetto esistente, puoi specificare l'uso della crittografia lato server con le AWS KMS keys (SSE-KMS) per crittografare i dati. Per fare ciò, aggiungi l'intestazione `x-amz-server-side-encryption` alla richiesta. Impostare il valore dell'intestazione sull'algoritmo di crittografia `aws:kms`. Amazon S3 conferma che

l'oggetto è stato archiviato utilizzando SSE-KMS restituendo l'intestazione della risposta `x-amz-server-side-encryption`.

Se specifichi l'intestazione `x-amz-server-side-encryption` con il valore `aws:kms`, puoi anche utilizzare le intestazioni di richiesta seguenti:

- `x-amz-server-side-encryption-aws-kms-key-id`
- `x-amz-server-side-encryption-context`
- `x-amz-server-side-encryption-bucket-key-enabled`

Argomenti

- Operazioni REST API di Amazon S3 che supportano SSE-KMS
- Contesto di crittografia (x-amz-server-side-encryption-context)
- AWS KMS ID chiave (x-amz-server-side-encryption-aws-kms-key-id)
- Chiavi bucket S3 (x-amz-server-side-encryption-aws-bucket-key-enabled)

Operazioni REST API di Amazon S3 che supportano SSE-KMS

Le operazioni REST API seguenti accettano le intestazioni di richiesta `x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-id` e `x-amz-server-side-encryption-context`.

- PutObject: quando carichi i dati utilizzando l'operazione API PUT, è possibile specificare queste intestazioni di richiesta.
- CopyObject: quando copi un oggetto, disponi di un oggetto di origine e un oggetto di destinazione. Quando si passano le intestazioni SSE-KMS con l'`CopyObject`operazione, queste vengono applicate solo all'oggetto di destinazione. Quando si copia un oggetto esistente, indipendentemente dal fatto che l'oggetto di origine sia crittografato o meno, l'oggetto di destinazione non viene crittografato a meno che non si richieda esplicitamente la crittografia lato server.
- POST Object— Quando si utilizza un'`POST`operazione per caricare un oggetto, anziché le intestazioni della richiesta, si forniscono le stesse informazioni nei campi del modulo.
- CreateMultipartUpload— Quando si caricano oggetti di grandi dimensioni utilizzando l'operazione API di caricamento multiparte, è possibile specificare queste intestazioni. Queste intestazioni vengono specificate nella richiesta. `CreateMultipartUpload`

Quando un oggetto viene archiviato utilizzando la crittografia lato server, le intestazioni di risposta delle seguenti operazioni REST API restituiscono l'intestazione `x-amz-server-side-encryption`.

- PutObject
- CopyObject
- POST Object
- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- GetObject
- HeadObject

> **Important**
>
> - Tutte `GET` le `PUT` richieste relative a un oggetto protetto da hanno AWS KMS esito negativo se non si effettuano tali richieste utilizzando Secure Sockets Layer (SSL), Transport Layer Security (TLS) o Signature Version 4.
> - Se il tuo oggetto utilizza SSE-KMS, non inviare le intestazioni delle richieste di crittografia per `GET` richieste e `HEAD` richieste, altrimenti riceverai un errore HTTP 400. BadRequest

Contesto di crittografia (**`x-amz-server-side-encryption-context`**)

Se si specifica `x-amz-server-side-encryption:aws:kms`, l'API Amazon S3 supporta un contesto di crittografia con l'intestazione `x-amz-server-side-encryption-context`. Un contesto di crittografia è un set di coppie chiave-valore che possono contenere ulteriori informazioni contestuali sui dati.

Amazon S3 utilizza automaticamente l'oggetto o il bucket Amazon Resource Name (ARN) come coppia di contesto di crittografia. Se utilizzi SSE-KMS senza abilitare una chiave bucket S3, usa l'ARN dell'oggetto come contesto di crittografia, ad esempio `arn:aws:s3:::`*`object_ARN`*. Se invece utilizzi SSE-KMS e abiliti una chiave bucket S3, usa l'ARN del bucket per il contesto di crittografia, ad esempio `arn:aws:s3:::`*`bucket_ARN`*.

Facoltativamente, è possibile fornire una coppia di contesto di crittografia aggiuntiva utilizzando l'intestazione `x-amz-server-side-encryption-context`. Tuttavia, poiché il contesto di crittografia non è crittografato, assicurati che non includa informazioni sensibili. Amazon S3 archivia questa coppia di chiavi aggiuntiva insieme al contesto di crittografia predefinito.

Per informazioni sul contesto di crittografia in Amazon S3, consulta la sezione Contesto di crittografia. Per informazioni generali sul contesto di crittografia, consulta Concetti di AWS Key Management Service : Contesto di crittografia nella Guida per gli sviluppatori di AWS Key Management Service .

AWS KMS ID chiave (**`x-amz-server-side-encryption-aws-kms-key-id`**)

Puoi utilizzare l'intestazione `x-amz-server-side-encryption-aws-kms-key-id` per specificare l'ID della chiave gestita dal cliente utilizzata per proteggere i dati. Se specifichi l'intestazione `x-amz-server-side-encryption:aws:kms`, ma non fornisci l'intestazione `x-amz-server-side-encryption-aws-kms-key-id`, Amazon S3 utilizza la Chiave gestita da AWS (`aws/s3`) per proteggere i dati. Se desideri utilizzare una chiave gestita dal cliente, devi fornire l'intestazione `x-amz-server-side-encryption-aws-kms-key-id` della chiave gestita dal cliente.

> **Important**
>
> Quando utilizzi una chiave KMS AWS KMS key per la crittografia lato server in Amazon S3, devi scegliere una chiave KMS di crittografia simmetrica. Amazon S3 supporta solo chiavi KMS di crittografia simmetrica. Per ulteriori informazioni sulle chiavi, consulta Chiavi KMS di crittografia simmetrica nella Guida per gli sviluppatori di AWS Key Management Service .

Chiavi bucket S3 (**`x-amz-server-side-encryption-aws-bucket-key-enabled`**)

Puoi utilizzare l'intestazione della `x-amz-server-side-encryption-aws-bucket-key-enabled` richiesta per abilitare o disabilitare una S3 Bucket Key a livello di oggetto. S3 Bucket Keys riduce i costi delle AWS KMS richieste diminuendo il traffico delle richieste da Amazon S3 a. AWS KMS Per ulteriori informazioni, consulta Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3.

Se specifichi l'intestazione `x-amz-server-side-encryption:aws:kms` ma non fornisci l'intestazione `x-amz-server-side-encryption-aws-bucket-key-enabled`, per crittografare l'oggetto saranno utilizzate le impostazioni della chiave bucket S3 per il bucket di destinazione. Per ulteriori informazioni, consulta Configurazione di una chiave bucket S3 a livello di oggetto .

Usando il AWS CLI

Per utilizzare i seguenti AWS CLI comandi di esempio, *`user input placeholders`* sostituiscili con le tue informazioni.

Quando caricate un nuovo oggetto o copiate un oggetto esistente, potete specificare l'uso della crittografia lato server con AWS KMS chiavi per crittografare i dati. Per fare ciò, aggiungi l'intestazione `--server-side-encryption aws:kms` alla richiesta. Utilizza il `--ssekms-key-id` *`example-key-id`* per aggiungere la AWS KMS chiave gestita dal cliente che hai creato. Se specifichi`--server-side-encryption aws:kms`, ma non fornisci un ID di AWS KMS chiave, Amazon S3 utilizzerà una chiave AWS gestita.

```
aws s3api put-object --bucket example-s3-bucket --key example-object-key --server-side-
encryption aws:kms --ssekms-key-id example-key-id --ssekms-encryption-context example-
encryption-context --body filepath
```

Puoi abilitare o disabilitare S3 Bucket Keys sulle tue `copy-object` operazioni `put-object` OO aggiungendo o. `--bucket-key-enabled` `--no-bucket-key-enabled` S3 Bucket Keys può ridurre i costi delle AWS KMS richieste diminuendo il traffico delle richieste da Amazon S3 a. AWS KMS Per ulteriori informazioni, consulta Ridurre il costo di SSE-KMS con S3 Bucket Keys.

```
aws s3api put-object --bucket example-s3-bucket --key example-object-key --server-side-
encryption aws:kms --bucket-key-enabled --body filepath
```

È possibile copiare un oggetto da un bucket di origine a un nuovo bucket e specificare la crittografia SSE-KMS.

```
aws s3api copy-object --copy-source example-s3-bucket/example-object-key --
bucket example-s3-bucket2 --key example-object-key --server-side-encryption aws:kms --
sse-kms-key-id example-key-id --ssekms-encryption-context example-encryption-context
```

Utilizzo degli SDK AWS

Quando usi AWS gli SDK, puoi richiedere che Amazon S3 venga AWS KMS keys utilizzato per la crittografia lato server. Gli esempi seguenti mostrano come utilizzare SSE-KMS con gli SDK per Java e.NET. AWS Per informazioni su altri SDK, consulta Codice di esempio e librerie nel Developer Center. AWS

> **Important**
>
> Quando utilizzi una chiave KMS AWS KMS key per la crittografia lato server in Amazon S3, devi scegliere una chiave KMS di crittografia simmetrica. Amazon S3 supporta solo chiavi KMS di crittografia simmetrica. Per ulteriori informazioni sulle chiavi, consulta Chiavi KMS di crittografia simmetrica nella Guida per gli sviluppatori di AWS Key Management Service .

Operazione **`CopyObject`**

Quando copi gli oggetti, puoi aggiungere le stesse proprietà della richiesta (`ServerSideEncryptionMethod` e `ServerSideEncryptionKeyManagementServiceKeyId`) per richiedere che Amazon S3 utilizzi una AWS KMS key. Per ulteriori informazioni sulla copia di oggetti, consulta la sezione Copiare, spostare e rinominare oggetti.

Operazione **`PUT`**

Java

Quando carichi un oggetto utilizzando il AWS SDK for Java, puoi richiedere ad Amazon S3 di utilizzare AWS KMS key un oggetto aggiungendo `SSEAwsKeyManagementParams` la proprietà come mostrato nella seguente richiesta:

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,
   keyName, file).withSSEAwsKeyManagementParams(new SSEAwsKeyManagementParams());
```

In questo caso, Amazon S3 utilizza Chiave gestita da AWS ()`aws/s3`. Per ulteriori informazioni, consulta Utilizzo della crittografia lato server con chiavi (SSE-KMS) AWS KMS. Facoltativamente, puoi creare una chiave KMS di crittografia simmetrica e specificarla nella richiesta, come mostrato nell'esempio seguente:

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,
   keyName, file).withSSEAwsKeyManagementParams(new
 SSEAwsKeyManagementParams(keyID));
```

Per ulteriori informazioni sulla creazione di chiavi gestite dai clienti, consulta Programming the AWS KMS API nella Developer Guide.AWS Key Management Service

Per esempi di codice di utilizzo per il caricamento di un oggetto, consulta gli argomenti elencati di seguito. Per usare questi esempi dovrai aggiornare gli esempi di codice e fornire informazioni sulla crittografia come mostrato nel frammento di codice precedente.

- Per il caricamento di un oggetto in un'unica operazione, consulta Caricamento degli oggetti.
- Per i caricamenti in più parti che utilizzano le operazioni dell'API di caricamento multiparte di alto o basso livello, consulta. Caricamento di un oggetto utilizzando il caricamento in più parti

.NET

Quando carichi un oggetto utilizzando il AWS SDK for .NET, puoi richiedere ad Amazon S3 di utilizzare AWS KMS key un oggetto aggiungendo `ServerSideEncryptionMethod` la proprietà come mostrato nella seguente richiesta:

```
PutObjectRequest putRequest = new PutObjectRequest
 {
     BucketName = example-s3-bucket,
     Key = keyName,
     // other properties
     ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS
 };
```

In questo caso, Amazon S3 utilizza il. Chiave gestita da AWS Per ulteriori informazioni, consulta Utilizzo della crittografia lato server con chiavi (SSE-KMS) AWS KMS. Facoltativamente, puoi creare la tua chiave di crittografia simmetrica gestita dal cliente e specificarla nella richiesta, come mostrato nell'esempio seguente:

```
PutObjectRequest putRequest1 = new PutObjectRequest
{
  BucketName = example-s3-bucket,
  Key = keyName,
  // other properties
  ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS,
  ServerSideEncryptionKeyManagementServiceKeyId = keyId
};
```

Per ulteriori informazioni sulla creazione di chiavi gestite dal cliente, consulta Programming the AWS KMS API nella Developer Guide.AWS Key Management Service

Per esempi di codice di utilizzo per il caricamento di un oggetto, consulta gli argomenti elencati di seguito. Per usare questi esempi dovrai aggiornare gli esempi di codice e fornire informazioni sulla crittografia come mostrato nel frammento di codice precedente.

- Per il caricamento di un oggetto in un'unica operazione, consulta Caricamento degli oggetti.
- Per i caricamenti in più parti che utilizzano le operazioni dell'API di caricamento multiparte di alto o basso livello, consulta. Caricamento di un oggetto utilizzando il caricamento in più parti

URL prefirmati

Java

Quando crei un URL predefinito per un oggetto crittografato con un AWS KMS key, devi specificare esplicitamente la versione 4 di Signature, come illustrato nell'esempio seguente:

```
ClientConfiguration clientConfiguration = new ClientConfiguration();
clientConfiguration.setSignerOverride("AWSS3V4SignerType");
AmazonS3Client s3client = new AmazonS3Client(
        new ProfileCredentialsProvider(), clientConfiguration);
...
```

Per un esempio di codice, consulta Condivisione di oggetti mediante URL prefirmati.

.NET

Quando si crea un URL predefinito per un oggetto crittografato con un AWS KMS key, è necessario specificare in modo esplicito la versione 4 della firma, come illustrato nell'esempio seguente:

```
AWSConfigs.S3Config.UseSignatureVersion4 = true;
```

Per un esempio di codice, consulta Condivisione di oggetti mediante URL prefirmati.

Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3

Amazon S3 Bucket Keys riduce il costo della crittografia lato server di Amazon S3 con AWS Key Management Service chiavi () (SSE-KMS).AWS KMS L'utilizzo di una chiave a livello di bucket per SSE-KMS può ridurre i costi delle AWS KMS richieste fino al 99 percento diminuendo il traffico delle richieste da Amazon S3 a. AWS KMS Con pochi clic nella AWS Management Console e senza

alcuna modifica alle applicazioni client, potrai configurare il bucket in modo da utilizzare una chiave bucket S3 per la crittografia SSE-KMS sui nuovi oggetti.

> **Note**
>
> Le S3 Bucket Keys non sono supportate per la crittografia lato server a doppio livello con chiavi () (DSSE-KMS). AWS Key Management Service AWS KMS

Chiavi bucket S3 per SSE-KMS

I carichi di lavoro che accedono a milioni o miliardi di oggetti crittografati con SSE-KMS possono generare grandi volumi di richieste verso. AWS KMSQuando usi SSE-KMS per proteggere i tuoi dati senza una S3 Bucket Key, Amazon S3 utilizza una chiave dati individuale per ogni oggetto. AWS KMS In questo caso, Amazon S3 effettua una chiamata AWS KMS ogni volta che viene effettuata una richiesta su un oggetto crittografato con KMS. Per informazioni sul funzionamento di SSE-KMS, consulta Utilizzo della crittografia lato server con chiavi (SSE-KMS) AWS KMS.

Quando configuri il bucket per utilizzare una chiave S3 Bucket per SSE-KMS, AWS genera una chiave a livello di bucket di breve durata da, quindi la conserva temporaneamente in S3. AWS KMS Questa chiave a livello di bucket creerà chiavi di dati per i nuovi oggetti durante il relativo ciclo di vita. Le S3 Bucket Key vengono utilizzate per un periodo di tempo limitato all'interno di Amazon S3, riducendo la necessità per S3 di effettuare richieste AWS KMS per completare le operazioni di crittografia. Ciò riduce il traffico da S3 a AWS KMS, consentendoti di accedere AWS KMS agli oggetti crittografati in Amazon S3 a una frazione del costo precedente.

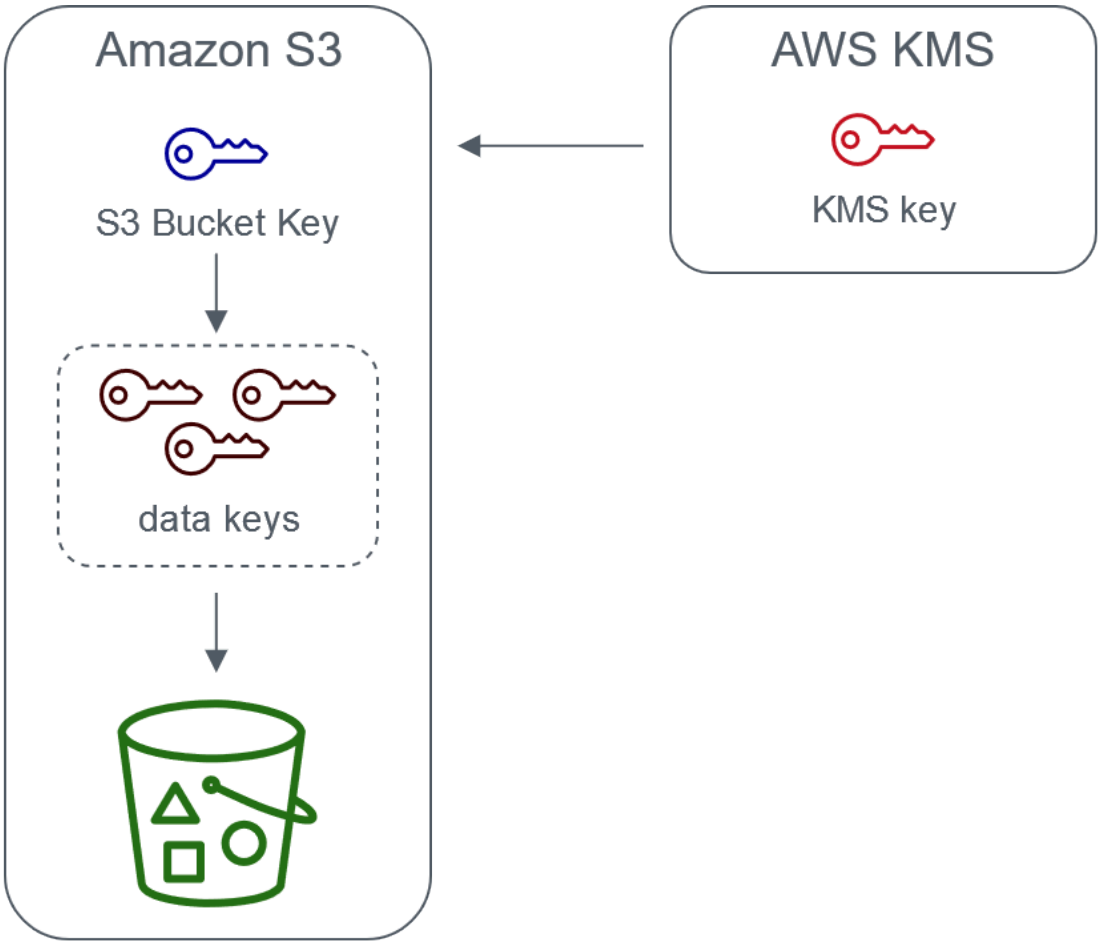
Le chiavi univoche a livello di bucket vengono recuperate almeno una volta per richiedente per garantire che l'accesso del richiedente alla chiave venga acquisito in un evento. AWS KMS CloudTrail Amazon S3 tratta i chiamanti come richiedenti diversi quando utilizzano ruoli o account diversi o lo stesso ruolo con politiche di ambito diverse. AWS KMS i risparmi sulle richieste riflettono il numero di richiedenti, i modelli di richiesta e l'età relativa degli oggetti richiesti. Ad esempio, un numero inferiore di richiedenti, la richiesta di più oggetti in una finestra temporale limitata e la crittografia con la stessa chiave a livello di bucket comportano un risparmio maggiore.

> **Note**
>
> L'utilizzo di S3 Bucket Keys ti consente di risparmiare sui costi delle AWS KMS richieste diminuendo le richieste a AWS KMS for `Encrypt` e le `Decrypt` operazioni tramite l'uso di

una chiave a livello di bucket. `GenerateDataKey` In base alla progettazione, le richieste successive che sfruttano questa chiave a livello di bucket non generano richieste AWS KMS API né convalidano l'accesso in base alla policy della chiave. AWS KMS

Quando si configura una chiave bucket S3, gli oggetti già presenti nel bucket non utilizzano la chiave Bucket S3. Per configurare una chiave bucket S3 per gli oggetti esistenti, è possibile utilizzare un'operazione `CopyObject`. Per ulteriori informazioni, consulta Configurazione di una chiave bucket S3 a livello di oggetto .

Amazon S3 condividerà una chiave S3 bucket solo per gli oggetti crittografati dalla stessa AWS KMS key. Le S3 Bucket Keys sono compatibili con le chiavi KMS create da AWS KMS, il materiale chiave importato e il materiale chiave supportato da archivi di chiavi personalizzati.



Server-side encryption with AWS Key Management service using an S3 Bucket Key

Configurazione delle chiavi bucket S3

Puoi configurare il tuo bucket per utilizzare una chiave S3 Bucket per SSE-KMS su nuovi oggetti tramite la console Amazon S3, gli SDK o l'API REST. AWS AWS CLI Con le chiavi di bucket S3 abilitate sul bucket, gli oggetti caricati con una chiave SSE-KMS specificata diversamente utilizzeranno chiavi di bucket S3 proprie. Indipendentemente dall'impostazione della chiave di bucket S3, puoi includere l'intestazione `x-amz-server-side-encryption-bucket-key-enabled`con un valore `true` o `false` or nella richiesta, per sovrascrivere l'impostazione del bucket.

Prima di configurare il bucket per utilizzare una chiave bucket S3, consulta Modifiche alla nota prima dell'abilitazione di una chiave bucket S3.

Configurazione di una chiave bucket S3 tramite la console di Amazon S3

Quando crei un nuovo bucket, puoi configurarlo in modo da utilizzare una chiave bucket S3 per SSE-KMS su nuovi oggetti. Puoi inoltre configurare un bucket esistente in modo utilizzare una chiave bucket S3 per SSE-KMS su nuovi oggetti aggiornando le proprietà del bucket.

Per ulteriori informazioni, consulta Configurazione del bucket per utilizzare una chiave bucket S3 con SSE-KMS per nuovi oggetti.

API REST e supporto SDK per S3 Bucket AWS CLI Keys AWS

Puoi utilizzare l'API REST o l' AWS SDK per configurare il tuo bucket in modo che utilizzi una S3 Bucket Key per SSE-KMS su nuovi oggetti. AWS CLI Puoi inoltre abilitare una chiave bucket S3 a livello di oggetto.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Configurazione di una chiave bucket S3 a livello di oggetto
- Configurazione del bucket per utilizzare una chiave bucket S3 con SSE-KMS per nuovi oggetti

Le seguenti operazioni API supportano le chiavi bucket S3 per SSE-KMS:

- PutBucketEncryption
  - `ServerSideEncryptionRule` accetta il parametro `BucketKeyEnabled` per abilitare e disabilitare una chiave bucket S3.
- GetBucketEncryption
  - `ServerSideEncryptionRule` restituisce le impostazioni per `BucketKeyEnabled`.

- PutObject,, e oggetto POST CopyObjectCreateMultipartUpload
  - L'intestazione della richiesta `x-amz-server-side-encryption-bucket-key-enabled` abilita o disabilita una chiave bucket S3 a livello di oggetto.
- HeadObject, GetObjectUploadPartCopy, UploadPart, e CompleteMultipartUpload
  - L'intestazione della risposta `x-amz-server-side-encryption-bucket-key-enabled` indica se una chiave bucket S3 è abilitata o disabilitata per un oggetto.

Lavorare con AWS CloudFormation

In AWS CloudFormation, la `AWS::S3::Bucket` risorsa include una proprietà di crittografia denominata `BucketKeyEnabled` che puoi utilizzare per abilitare o disabilitare una S3 Bucket Key.

Per ulteriori informazioni, consulta Usando AWS CloudFormation.

Modifiche alla nota prima dell'abilitazione di una chiave bucket S3

Prima di abilitare una chiave bucket S3, tieni presente le seguenti modifiche correlate:

IAM o politiche chiave AWS KMS

Se le tue policy AWS Identity and Access Management (IAM) o le policy AWS KMS chiave esistenti utilizzano il tuo oggetto Amazon Resource Name (ARN) come contesto di crittografia per perfezionare o limitare l'accesso alla tua chiave KMS, queste policy non funzioneranno con una S3 Bucket Key. Le chiavi bucket S3 utilizzano l'ARN del bucket come contesto di crittografia. Prima di abilitare una chiave S3 Bucket, aggiorna le policy IAM o le policy AWS KMS chiave per utilizzare l'ARN del bucket come contesto di crittografia.

Per ulteriori informazioni sul contesto di crittografia e sulle chiavi bucket S3, consulta Contesto di crittografia.

CloudTrail eventi per AWS KMS

Dopo aver abilitato una S3 Bucket Key, AWS KMS CloudTrail gli eventi registrano l'ARN del bucket anziché l'ARN dell'oggetto. Inoltre, nei log vengono visualizzati meno CloudTrail eventi KMS per gli oggetti SSE-KMS. Poiché il materiale chiave è limitato nel tempo in Amazon S3, vengono inviate meno richieste. AWS KMS

Utilizzo di una chiave bucket S3 con la replica

Le chiavi bucket S3 possono essere utilizzate con la replica della stessa regione (SRR) e con la replica tra regioni (CRR).

Quando Amazon S3 replica un oggetto crittografato, in genere conserva le impostazioni di crittografia dell'oggetto di replica nel bucket di destinazione. Tuttavia, se l'oggetto di origine non è crittografato e il bucket di destinazione utilizza la crittografia predefinita o una chiave bucket S3, Amazon S3 crittografa l'oggetto con la configurazione del bucket di destinazione.

Negli esempi seguenti viene illustrato il funzionamento di una chiave bucket S3 con la replica. Per ulteriori informazioni, consulta Replica di oggetti crittografati (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

Example Esempio 1: l'oggetto di origine utilizza le chiavi bucket S3 e il bucket di destinazione usa la crittografia predefinita

Se l'oggetto di origine utilizza una chiave bucket S3 ma il bucket di destinazione utilizza la crittografia predefinita con SSE-KMS, l'oggetto di replica mantiene le impostazioni di crittografia della chiave bucket S3 nel bucket di destinazione. Il bucket di destinazione utilizza ancora la crittografia predefinita con SSE-KMS.

Example Esempio 2: l'oggetto di origine non è crittografato e il bucket di destinazione usa una chiave bucket S3 con SSE-KMS

Se l'oggetto di origine non è crittografato e il bucket di destinazione usa una chiave bucket S3 con SSE-KMS, l'oggetto di replica viene crittografato con una chiave bucket S3 utilizzando SSE-KMS nel bucket di destinazione. Ciò fa sì che l'`ETag` dell'oggetto di origine sia diverso dall'`ETag` dell'oggetto replicato. È necessario aggiornare le applicazioni che utilizzano l'`ETag` per tenere conto di tale differenza.

Operazioni con le chiavi bucket S3

Per ulteriori informazioni sull'abilitazione e l'utilizzo di chiavi bucket S3, consulta le sezioni seguenti:

- Configurazione del bucket per utilizzare una chiave bucket S3 con SSE-KMS per nuovi oggetti
- Configurazione di una chiave bucket S3 a livello di oggetto
- Visualizzazione delle impostazioni per una chiave bucket S3

Configurazione del bucket per utilizzare una chiave bucket S3 con SSE-KMS per nuovi oggetti

Quando configuri la crittografia lato server con le chiavi AWS Key Management Service (AWS KMS) (SSE-KMS), puoi configurare il bucket per utilizzare una S3 Bucket Key per SSE-KMS su nuovi

oggetti. Le S3 Bucket Keys riducono il traffico delle richieste da Amazon S3 AWS KMS a SSE-KMS e riducono il costo. Per ulteriori informazioni, consulta Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3.

Puoi configurare il tuo bucket per utilizzare una chiave S3 Bucket per SSE-KMS su nuovi oggetti utilizzando la console Amazon S3, l'API REST, gli SDK, () o. AWS AWS Command Line Interface AWS CLI AWS CloudFormation Se desideri abilitare o disabilitare una chiave bucket S3 per gli oggetti esistenti, puoi utilizzare un'operazione `CopyObject`. Per ulteriori informazioni, consulta Configurazione di una chiave bucket S3 a livello di oggetto e Utilizzo delle operazioni in batch S3 per crittografare oggetti con chiavi bucket S3.

Quando una chiave bucket S3 è abilitata per il bucket di origine o di destinazione, il contesto di crittografia sarà l'Amazon Resource Name (ARN) del bucket e non l'ARN dell'oggetto, ad esempio, `arn:aws:s3:::`*`bucket_ARN`*. Dovrai aggiornare le policy IAM per utilizzare l'ARN del bucket per il contesto di crittografia. Per ulteriori informazioni, consulta Chiavi bucket S3 e replica.

Negli esempi seguenti viene illustrato il funzionamento di una chiave bucket S3 con la replica. Per ulteriori informazioni, consulta Replica di oggetti crittografati (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

Prerequisiti

Prima di configurare il bucket per utilizzare una chiave bucket S3, consulta Modifiche alla nota prima dell'abilitazione di una chiave bucket S3.

Utilizzo della console S3

Nella console S3, puoi abilitare o disabilitare una chiave bucket S3 per un bucket nuovo o esistente. Gli oggetti nella console S3 ereditano l'impostazione della chiave bucket S3 dalla configurazione del bucket. Quando abiliti una chiave bucket S3 per il bucket, i nuovi oggetti caricati nel bucket utilizzano una chiave bucket S3 per SSE-KMS.

Caricamento, copia o modifica di oggetti nei bucket che dispongono di una chiave bucket S3 abilitata

Se carichi, modifichi o copi un oggetto in un bucket con una chiave bucket S3 abilitata, le impostazioni della chiave bucket S3 per tale oggetto potrebbero essere aggiornate in modo da allinearsi alla configurazione del bucket.

Se un oggetto ha già una chiave bucket S3 abilitata, le impostazioni della chiave bucket S3 per quell'oggetto non cambiano quando si copia o si modifica l'oggetto. Tuttavia, se modifichi o copi

un oggetto che non dispone di una chiave bucket S3 attivata e il bucket di destinazione ha una configurazione con una chiave bucket S3, l'oggetto eredita le impostazioni della chiave bucket S3 del bucket di destinazione. Ad esempio, se l'oggetto di origine non ha una chiave bucket S3 abilitata ma il bucket di destinazione ne ha una, una chiave bucket S3 è abilitata per l'oggetto.

Abilitazione di una chiave bucket S3 quando si crea un nuovo bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Seleziona Crea bucket.
4. Inserisci il nome del bucket e scegli la tua Regione AWS.
5. In Crittografia predefinita, scegli Chiave AWS Key Management Service (SSE-KMS) per Tipo di chiave di crittografia.
6. In Chiave AWS KMS effettua una delle seguenti operazioni per scegliere la chiave KMS:
   - Per scegliere da un elenco di chiavi KMS disponibili, scegli Scegli tra le tue AWS KMS keys, quindi scegli la tua chiave KMS dall'elenco delle chiavi disponibili.

     In questo elenco vengono visualizzate sia la chiave Chiave gestita da AWS (`aws/s3`) che quella gestita dai clienti. Per ulteriori informazioni sulle chiavi gestite dai clienti, consulta Customer keys and AWS keys nella AWS Key Management Service Developer Guide.
   - Per specificare l'ARN della chiave KMS, scegli Inserisci l'ARN della AWS KMS key e quindi specifica l'ARN della chiave KMS nel campo visualizzato.
   - Per creare una nuova chiave gestita dal cliente nella AWS KMS console, scegli Crea una chiave KMS.

     Per ulteriori informazioni sulla creazione di una AWS KMS key, consulta Creating Keys nella AWS Key Management Service Developer Guide.
7. In Chiave bucket scegli Abilita.
8. Scegliere Create bucket (Crea bucket).

   Amazon S3 crea il tuo bucket con una chiave bucket S3 abilitata. I nuovi oggetti caricati nel bucket utilizzeranno una chiave bucket S3.

   Per disabilitare una chiave bucket S3, completa i passaggi precedenti e scegli Disabilita.

Abilitazione di una chiave bucket S3 per un bucket esistente

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket scegli il bucket per cui desideri abilitare una chiave bucket S3.
4. Scegliere la scheda Properties (Proprietà).
5. In Default encryption (Crittografia di default), scegliere Edit (Modifica).
6. In Crittografia predefinita, scegli Chiave AWS Key Management Service (SSE-KMS) per Tipo di chiave di crittografia.
7. In Chiave AWS KMS effettua una delle seguenti operazioni per scegliere la chiave KMS:

   - Per scegliere da un elenco di chiavi KMS disponibili, scegli tra le tue chiavi KMS AWS KMS keys, quindi scegli la tua chiave KMS dall'elenco delle chiavi disponibili.

     In questo elenco vengono visualizzate sia la chiave Chiave gestita da AWS (`aws/s3`) che quella gestita dai clienti. Per ulteriori informazioni sulle chiavi gestite dai clienti, consulta Customer keys and AWS keys nella AWS Key Management Service Developer Guide.
   - Per specificare l'ARN della chiave KMS, scegli Inserisci l'ARN della AWS KMS key e quindi specifica l'ARN della chiave KMS nel campo visualizzato.
   - Per creare una nuova chiave gestita dal cliente nella AWS KMS console, scegli Crea una chiave KMS.

     Per ulteriori informazioni sulla creazione di una AWS KMS key, consulta Creating Keys nella AWS Key Management Service Developer Guide.
8. In Chiave bucket scegli Abilita.
9. Seleziona Salva modifiche.

Amazon S3 abilita una chiave bucket S3 per i nuovi oggetti aggiunti al tuo bucket. Gli oggetti esistenti non utilizzano la chiave bucket S3. Per configurare una chiave bucket S3 per gli oggetti esistenti, è possibile utilizzare un'operazione `CopyObject`. Per ulteriori informazioni, consulta Configurazione di una chiave bucket S3 a livello di oggetto .

Per disabilitare una chiave bucket S3, completa i passaggi precedenti e scegli Disabilita.

Utilizzo dell'API REST

Puoi utilizzarla PutBucketEncryptionper abilitare o disabilitare una S3 Bucket Key per il tuo bucket. Per configurare una S3 Bucket Key con`PutBucketEncryption`, usa il tipo di ServerSideEncryptionRuledati, che include la crittografia predefinita con SSE-KMS. Puoi inoltre utilizzare una chiave gestita dal cliente specificando l'ID della chiave KMS per la chiave gestita dal cliente.

Per ulteriori informazioni ed esempi di sintassi, consulta. PutBucketEncryption

Utilizzo dell' AWS SDK for Java

Nell'esempio seguente viene abilitata la crittografia bucket predefinita con SSE-KMS e una chiave bucket S3 utilizzando la 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);
```

Usando il AWS CLI

Nell'esempio seguente viene abilitata la crittografia bucket predefinita con SSE-KMS e una chiave bucket S3 utilizzando la AWS CLI. Sostituire *`user input placeholders`* con le proprie informazioni.

```
aws s3api put-bucket-encryption --bucket example-s3-bucket --server-side-encryption-
configuration '{
        "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "aws:kms",
                    "KMSMasterKeyID": "KMS-Key-ARN"
                },
                "BucketKeyEnabled": true
            }
        ]
    }'
```

Usando AWS CloudFormation

Per ulteriori informazioni sulla configurazione di una S3 Bucket Key con AWS CloudFormation, consulta AWS::S3::Bucket ServerSideEncryptionRulela Guida per l'AWS CloudFormation utente.

Configurazione di una chiave bucket S3 a livello di oggetto

Quando esegui un'operazione PUT o COPY utilizzando l'API REST, AWS gli SDK o AWS CLI, puoi abilitare o disabilitare una S3 Bucket Key a livello di oggetto aggiungendo l'intestazione della `x-amz-server-side-encryption-bucket-key-enabled` richiesta con un valore or. `true` `false` S3 Bucket Keys riduce il costo della crittografia lato server utilizzando AWS Key Management Service (AWS KMS) (SSE-KMS) diminuendo il traffico delle richieste da Amazon S3 a. AWS KMS Per ulteriori informazioni, consulta Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3.

Quando configuri una chiave bucket S3 per un oggetto utilizzando un'operazione PUT o COPY, Amazon S3 aggiorna le impostazioni solo per quell'oggetto. Le impostazioni della chiave bucket S3 per il bucket di destinazione non cambiano. Se invii una richiesta PUT o COPY per un oggetto crittografato con KMS in un bucket con le chiavi di bucket S3 abilitate, l'operazione a livello di oggetto utilizzerà automaticamente le chiavi di bucket S3 a meno che non disabiliti le chiavi nell'intestazione della richiesta. Se non specifichi una chiave bucket S3 per il tuo oggetto, Amazon S3 applica le impostazioni della chiave bucket S3 per il bucket di destinazione all'oggetto.

Prerequisito

Prima di configurare l'oggetto per utilizzare una chiave bucket S3, consulta Modifiche alla nota prima dell'abilitazione di una chiave bucket S3.

Argomenti

- Operazioni in Batch Amazon S3
- Utilizzo di REST API
- Utilizzo dell' AWS SDK per Java PutObject ()
- Usando il AWS CLI () PutObject

Operazioni in Batch Amazon S3

Per crittografare gli oggetti Amazon S3 esistenti, puoi utilizzare le operazioni in batch di Amazon S3. Fornisci alle operazioni in batch S3 un elenco di oggetti da utilizzare e le operazioni in batch chiamano la rispettiva API per eseguire l'operazione specifica.

Puoi utilizzare l'operazione di copia delle operazioni in batch S3 per copiare gli oggetti non crittografati esistenti e scriverli nello stesso bucket degli oggetti crittografati. Un solo processo di operazioni in batch può eseguire l'operazione specificata su miliardi di oggetti. Per ulteriori informazioni, consulta Esecuzione di operazioni in batch su larga scala su oggetti Amazon S3 e Crittografia di oggetti con le operazioni in batch di Amazon S3.

Utilizzo di REST API

Quando utilizzi SSE-KMS, puoi abilitare una chiave bucket S3 per un oggetto utilizzando le seguenti operazioni API:

- PutObject— Quando carichi un oggetto, puoi specificare l'intestazione della `x-amz-server-side-encryption-bucket-key-enabled` richiesta per abilitare o disabilitare una S3 Bucket Key a livello di oggetto.
- CopyObject— Quando copi un oggetto e configuri SSE-KMS, puoi specificare l'intestazione della `x-amz-server-side-encryption-bucket-key-enabled` richiesta per abilitare o disabilitare una S3 Bucket Key per il tuo oggetto.
- POST Object: quando esegui un'operazione `POST` per caricare un oggetto e configurare SSE-KMS, puoi utilizzare il campo del modulo `x-amz-server-side-encryption-bucket-key-enabled` per abilitare o disabilitare una chiave bucket S3 per l'oggetto.
- CreateMultipartUpload— Quando carichi oggetti di grandi dimensioni utilizzando l'operazione `CreateMultipartUpload` API e configuri SSE-KMS, puoi utilizzare l'intestazione della `x-amz-server-side-encryption-bucket-key-enabled` richiesta per abilitare o disabilitare una S3 Bucket Key per il tuo oggetto.

Per abilitare una chiave bucket S3 a livello di oggetto, dovrai includere l'intestazione della richiesta `x-amz-server-side-encryption-bucket-key-enabled`. Per ulteriori informazioni su SSE-KMS e REST API, consulta la sezione Utilizzo di REST API.

Utilizzo dell' AWS SDK per Java PutObject ()

Il seguente esempio può essere utilizzato per configurare una chiave bucket S3 a livello di oggetto utilizzando AWS SDK for Java.

Java

```
AmazonS3 s3client = AmazonS3ClientBuilder.standard()
    .withRegion(Regions.DEFAULT_REGION)
    .build();

String bucketName = "DOC-EXAMPLE-BUCKET1";
String keyName = "key name for object";
String contents = "file contents";

PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, keyName,
 contents)
    .withBucketKeyEnabled(true);

s3client.putObject(putObjectRequest);
```

Usando il AWS CLI () PutObject

È possibile utilizzare il seguente AWS CLI esempio per configurare una S3 Bucket Key a livello di oggetto come parte di una `PutObject` richiesta.

```
aws s3api put-object --bucket example-s3-bucket --key object key name --server-side-
encryption aws:kms --bucket-key-enabled --body filepath
```

Visualizzazione delle impostazioni per una chiave bucket S3

Puoi visualizzare le impostazioni per una chiave S3 Bucket a livello di bucket o oggetto utilizzando la console Amazon S3, l'API REST AWS Command Line Interface ()AWS CLI o gli SDK. AWS

Le S3 Bucket Keys riducono il traffico delle richieste da Amazon S3 AWS KMS a (SSE-KMS) e riducono il costo della crittografia lato server. AWS Key Management Service Per ulteriori informazioni, consulta Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3.

Per visualizzare le impostazioni della chiave bucket S3 per un bucket o un oggetto che ha ereditato le impostazioni della chiave bucket S3 dalla configurazione del bucket, è necessaria l'autorizzazione per eseguire l'operazione `s3:GetEncryptionConfiguration`. Per ulteriori informazioni, consulta il riferimento GetBucketEncryptionall'API di Amazon Simple Storage Service.

Utilizzo della console S3

Nella console S3, puoi visualizzare le impostazioni della chiave bucket S3 per il bucket o l'oggetto. Le impostazioni della chiave bucket S3 vengono ereditate dalla configurazione del bucket a meno che gli oggetti di origine non dispongano già di una chiave bucket S3 configurata.

Oggetti e cartelle nello stesso bucket possono avere diverse impostazioni della chiave bucket S3. Ad esempio, se carichi un oggetto utilizzando REST API e abiliti una chiave bucket S3 per tale oggetto, questo manterrà l'impostazione della chiave bucket S3 nel bucket di destinazione anche se la chiave bucket S3 è disabilitata. Come altro esempio, se abiliti una chiave bucket S3 per un bucket esistente, gli oggetti già presenti nel bucket non utilizzeranno una chiave bucket S3. Tuttavia, i nuovi oggetti avranno una chiave bucket S3 abilitata.

Visualizzazione dell'impostazione della chiave bucket S3 per il bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket scegli il bucket per cui desideri abilitare una chiave bucket S3.
4. Scegliere Properties (Proprietà).
5. Nella sezione Crittografia predefinita, in Chiave bucket, viene visualizzata l'impostazione della chiave bucket S3 per il bucket.

   Se non riesci a visualizzare l'impostazione della chiave bucket S3, è possibile che non disponi dell'autorizzazione per eseguire l'operazione `s3:GetEncryptionConfiguration`. Per ulteriori informazioni, consulta il riferimento GetBucketEncryptionall'API di Amazon Simple Storage Service.

Visualizzazione dell'impostazione della chiave bucket S3 per l'oggetto

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket scegli il bucket per cui desideri abilitare una chiave bucket S3.
3. Nell'elenco Oggetti scegli il nome dell'oggetto.
4. Nella scheda Dettagli , in Impostazioni di crittografia lato server, seleziona Modifica.

   In Chiave bucket è visualizzata l'impostazione della chiave bucket S3 per l'oggetto. Non è possibile modificare questa impostazione.

Usando il AWS CLI

Restituzione delle impostazioni della chiave bucket S3 a livello di bucket

Per usare questo comando, sostituire *user input placeholder* con le proprie informazioni.

```
aws s3api get-bucket-encryption --bucket example-s3-bucket1
```

Per ulteriori informazioni, vedere get-bucket-encryptionnel AWS CLI Command Reference.

Restituzione delle impostazioni a livello di oggetto di una chiave bucket S3

Per usare questo comando, sostituire *user input placeholder* con le proprie informazioni.

```
aws s3api head-object --bucket example-s3-bucket1 --key my_images.tar.bz2
```

Per ulteriori informazioni, consulta head-object in Guida di riferimento dei comandi di AWS CLI .

Utilizzo di REST API

Restituzione delle impostazioni della chiave bucket S3 a livello di bucket

Per restituire le informazioni di crittografia per un bucket, incluse le impostazioni per una chiave bucket S3, utilizza l'operazione `GetBucketEncryption`. Le impostazioni della chiave bucket S3 vengono restituite nel corpo della risposta nell'elemento `ServerSideEncryptionConfiguration` con l'impostazione `BucketKeyEnabled`. Per ulteriori informazioni, GetBucketEncryptionconsulta Amazon S3 API Reference.

Restituzione delle impostazioni a livello di oggetto per una chiave bucket S3

Per restituire lo stato della chiave bucket S3 per un oggetto, utilizza l'operazione `HeadObject`. `HeadObject` restituisce l'intestazione della risposta `x-amz-server-side-encryption-bucket-key-enabled` per mostrare se una chiave bucket S3 è abilitata o disabilitata per l'oggetto. Per ulteriori informazioni, HeadObjectconsulta Amazon S3 API Reference.

Le seguenti operazioni delle API restituiscono l'intestazione della risposta `x-amz-server-side-encryption-bucket-key-enabled` anche se una chiave bucket S3 è configurata per un oggetto:

- PutObject
- PostObject
- CopyObject
- CreateMultipartUpload
- UploadPartCopy
- UploadPart
- CompleteMultipartUpload
- GetObject

Utilizzo della crittografia lato server a doppio livello con chiavi (DSSE-KMS) AWS KMS

L'utilizzo della crittografia lato server a due livelli con AWS Key Management Service (AWS KMS) chiavi (DSSE-KMS) applica due livelli di crittografia agli oggetti quando vengono caricati su Amazon S3. DSSE-KMS consente di soddisfare più facilmente gli standard di conformità che richiedono l'applicazione della crittografia a più livelli ai dati e il pieno controllo delle chiavi di crittografia.

Quando usi DSSE-KMS con un bucket Amazon S3, AWS KMS le chiavi devono trovarsi nella stessa regione del bucket. Inoltre, quando per l'oggetto è richiesta la crittografia DSSE-KMS, il checksum S3 come parte dei metadati dell'oggetto viene archiviato in formato crittografato. Per ulteriori informazioni sui checksum, consulta Verifica dell'integrità degli oggetti.

Sono previsti costi aggiuntivi per l'utilizzo di DSSE-KMS e. AWS KMS keys Per ulteriori informazioni sui prezzi di DSSE-KMS, consulta Concetti di AWS KMS key nella Guida per gli sviluppatori di AWS Key Management Service e Prezzi di AWS KMS.

> **Note**
>
> Le chiavi bucket S3 non sono supportate per DSSE-KMS.

Richiede la crittografia lato server a doppio livello con (DSSE-KMS) AWS KMS keys

Per richiedere la crittografia lato server a doppio livello di tutti gli oggetti in uno specifico bucket Amazon S3, è possibile utilizzare una policy del bucket. Ad esempio, la seguente policy del bucket rifiuta a chiunque l'autorizzazione al caricamento dell'oggetto (`s3:PutObject`) se la richiesta non include un'intestazione `x-amz-server-side-encryption` che richiede la crittografia lato server con DSSE-KMS.

```
{
            "Version":"2012-10-17",
            "Id":"PutObjectPolicy",
            "Statement":[{
                  "Sid":"DenyUnEncryptedObjectUploads",
                  "Effect":"Deny",
                  "Principal":"*",
                  "Action":"s3:PutObject",
                  "Resource":"arn:aws:s3:::example-s3-bucket1/*",
                  "Condition":{
                     "StringNotEquals":{
                        "s3:x-amz-server-side-encryption":"aws:kms:dsse"
                     }
                  }
               }
            ]
         }
```

Argomenti

- Specifica della crittografia lato server a doppio livello con chiavi AWS KMS (DSSE-KMS)

Specifica della crittografia lato server a doppio livello con chiavi AWS KMS (DSSE-KMS)

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon

S3 aggiuntiva negli SDK and. AWS Command Line Interface AWS Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

Tutti i bucket Amazon S3 hanno la crittografia configurata per impostazione predefinita e tutti i nuovi oggetti caricati in un bucket S3 vengono automaticamente crittografati quando sono a riposo. La crittografia lato server con le chiavi gestite da Amazon S3 (SSE-S3) è la configurazione predefinita della crittografia per ogni bucket di Amazon S3. Per utilizzare un diverso tipo di crittografia, puoi specificare il tipo di crittografia lato server da utilizzare nelle richieste PUT S3 oppure impostare la configurazione di crittografia predefinita nel bucket di destinazione.

Se desideri specificare un tipo di crittografia diverso nelle tue PUT richieste, puoi utilizzare la crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS), la crittografia lato server a due livelli con AWS KMS chiavi (DSSE-KMS) o la crittografia lato server con chiavi fornite dal cliente (SSE-C). Per impostare una configurazione di crittografia predefinita diversa nel bucket di destinazione puoi utilizzare SSE-KMS o DSSE-KMS.

È possibile applicare la crittografia quando stai caricando un nuovo oggetto o copiando un oggetto esistente.

È possibile specificare DSSE-KMS utilizzando la console Amazon S3, la REST API di Amazon S3 e la AWS Command Line Interface (AWS CLI). Per ulteriori informazioni, consulta i seguenti argomenti.

Note

Puoi usare più regioni AWS KMS keys in Amazon S3. Tuttavia, Amazon S3 attualmente tratta le chiavi multi-regione come se fossero chiavi a regione singola e non utilizza le caratteristiche multi-regione della chiave. Per ulteriori informazioni, consulta Utilizzo delle chiavi multi-regione nella Guida per gli sviluppatori di AWS Key Management Service .

Note

Se desideri utilizzare una chiave KMS di proprietà di un account diverso, devi avere l'autorizzazione necessaria per l'uso della chiave. Per ulteriori informazioni sulle autorizzazioni tra account per le chiavi KMS, vedi Creazione di chiavi KMS utilizzabili da altri account nella Guida per gli sviluppatori di AWS Key Management Service .

Utilizzo della console S3

Questa sezione descrive come impostare o modificare il tipo di crittografia di un oggetto per utilizzare la crittografia lato server a doppio livello con AWS Key Management Service (AWS KMS) chiavi (DSSE-KMS) utilizzando la console Amazon S3.

> **Note**
>
> - Se si modifica il metodo di crittografia di un oggetto, viene creato un nuovo oggetto per sostituire quello precedente. Se è abilitata la funzione Controllo delle versioni S3, viene creata una nuova versione dell'oggetto e l'oggetto esistente diventa una versione precedente. Il ruolo che modifica la proprietà diventa anche il proprietario del nuovo oggetto o della versione dell'oggetto.
> - Se modifichi il tipo di crittografia per un oggetto con tag definiti dall'utente, devi disporre dell'autorizzazione. `s3:GetObjectTagging` Se state modificando il tipo di crittografia per un oggetto che non ha tag definiti dall'utente ma ha una dimensione superiore a 16 MB, è necessario disporre anche dell'`s3:GetObjectTagging`autorizzazione.
>
> Se la policy del bucket di destinazione nega l'`s3:GetObjectTagging`azione, il tipo di crittografia dell'oggetto verrà aggiornato, ma i tag definiti dall'utente verranno rimossi dall'oggetto e verrà visualizzato un errore.

Per aggiungere o modificare la crittografia di un oggetto

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket scegli il nome del bucket contenente gli oggetti da crittografare.
4. Nell'elenco Oggetti seleziona la casella di controllo accanto al nome dell'oggetto per cui vuoi aggiungere o modificare la crittografia.

   Viene visualizzata la pagina dei dettagli dell'oggetto, con diverse sezioni che visualizzano le proprietà dell'oggetto.
5. Scegliere la scheda Properties (Proprietà).
6. Scorri verso il basso fino alla sezione Crittografia predefinita e scegli Modifica.

Viene visualizzata la pagina Modifica la crittografia predefinita.

7. In Tipo di crittografia, scegli Crittografia lato server a doppio livello con AWS Key Management Service chiavi (DSSE-KMS).
8. In Chiave AWS KMS effettua una delle seguenti operazioni per scegliere la chiave KMS:

   - Per scegliere da un elenco di chiavi KMS disponibili, seleziona Scegli tra le chiavi AWS KMS keys, quindi scegli la chiave KMS dall'elenco delle chiavi disponibili.

     In questo elenco vengono visualizzate sia la Chiave gestita da AWS chiave (`aws/s3`) che quella gestita dal cliente. Per ulteriori informazioni sulle chiavi gestite dal cliente, consulta Chiavi gestite dal cliente e chiavi AWS nella Guida per gli sviluppatori di AWS Key Management Service .
   - Per inserire l'ARN della chiave KMS, scegli Inserisci AWS KMS key ARN, quindi inserisci l'ARN della chiave KMS nel campo visualizzato.
   - Per creare una nuova chiave gestita dal cliente nella AWS KMS console, scegli Crea una chiave KMS.

     Per ulteriori informazioni sulla creazione di una AWS KMS key, consulta Creating keys nella AWS Key Management Service Developer Guide.

   > **Important**
   >
   > Puoi utilizzare solo le chiavi KMS disponibili nella stessa Regione AWS del bucket. La console Amazon S3 elenca solo le prime 100 chiavi KMS nella stessa regione del bucket. Per utilizzare una chiave KMS non elencata, devi inserire l'ARN della chiave KMS. Se desideri utilizzare una chiave KMS di proprietà di un account diverso, è necessario innanzitutto disporre dell'autorizzazione necessaria per l'uso della chiave e quindi inserire l'ARN della chiave KMS.
   > Amazon S3 supporta solo chiavi KMS di crittografia simmetriche e non chiavi KMS asimmetriche. Per ulteriori informazioni, consulta Identificazione delle chiavi KMS asimmetriche nella Guida per gli sviluppatori di AWS Key Management Service .

9. Per Chiave bucket scegli Disabilita. Le chiavi bucket S3 non sono supportate per DSSE-KMS.
10. Seleziona Salva modifiche.

> **Note**
>
> Questa azione applica la crittografia a tutti gli oggetti specificati. Durante la crittografia delle cartelle, attendere il completamento dell'operazione di salvataggio prima di aggiungere nuovi oggetti alla cartella.

Utilizzo di REST API

Quando si crea un oggetto, ovvero quando si carica un nuovo oggetto o si copia un oggetto esistente, è possibile specificare l'uso della crittografia lato server a doppio livello con AWS KMS keys (DSSE-KMS) per crittografare i dati. Per fare ciò, aggiungi l'intestazione `x-amz-server-side-encryption` alla richiesta. Impostare il valore dell'intestazione sull'algoritmo di crittografia `aws:kms:dsse`. Amazon S3 conferma che l'oggetto è stato archiviato utilizzando la crittografia DSSE-KMS restituendo l'intestazione della risposta `x-amz-server-side-encryption`.

Se specifichi l'intestazione `x-amz-server-side-encryption` con il valore `aws:kms:dsse`, puoi anche utilizzare le intestazioni di richiesta seguenti:

- `x-amz-server-side-encryption-aws-kms-key-id: SSEKMSKeyId`
- `x-amz-server-side-encryption-context: SSEKMSEncryptionContext`

Argomenti

- Operazioni REST API di Amazon S3 che supportano DSSE-KMS
- Contesto di crittografia (x-amz-server-side-encryption-context)
- AWS KMS ID chiave () x-amz-server-side-encryption-aws-kms-key-id

Operazioni REST API di Amazon S3 che supportano DSSE-KMS

Le operazioni REST API seguenti accettano le intestazioni di richiesta `x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-id` e `x-amz-server-side-encryption-context`.

- PutObject: quando carichi i dati utilizzando l'operazione API PUT, è possibile specificare queste intestazioni di richiesta.
- CopyObject: quando copi un oggetto, disponi di un oggetto di origine e un oggetto di destinazione. Tuttavia, le intestazioni DSSE-KMS passate con l'operazione `CopyObject` vengono applicate solo

all'oggetto di destinazione. Quando si copia un oggetto esistente, indipendentemente dal fatto che l'oggetto di origine sia stato o meno crittografato, l'oggetto di destinazione non viene crittografato, a meno che non si richieda esplicitamente la crittografia lato server.

- POST Object: quando utilizzi un'operazione `POST` per caricare un oggetto, anziché le intestazioni di richiesta, specifica le stesse informazioni dei campi del modulo.
- CreateMultipartUpload: quando carichi oggetti di grandi dimensioni utilizzando il caricamento in più parti, puoi specificare queste intestazioni nella richiesta `CreateMultipartUpload`.

Quando un oggetto viene archiviato con la crittografia lato server, le intestazioni di risposta delle seguenti operazioni REST API restituiscono l'intestazione `x-amz-server-side-encryption`.

- PutObject
- CopyObject
- POST Object
- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- GetObject
- HeadObject

> **Important**
>
> - Tutte `GET` le `PUT` richieste relative a un oggetto protetto da hanno AWS KMS esito negativo se non vengono effettuate utilizzando Secure Sockets Layer (SSL), Transport Layer Security (TLS) o Signature Version 4.
> - Se l'oggetto utilizza DSSE-KMS, non inviare intestazioni di richiesta di crittografia per le richieste `GET` e `HEAD` per evitare di ricevere un errore HTTP 400 (richiesta non valida).

Contesto di crittografia (**`x-amz-server-side-encryption-context`**)

Se si specifica `x-amz-server-side-encryption:aws:kms:dsse`, l'API Amazon S3 supporta un contesto di crittografia con l'intestazione `x-amz-server-side-encryption-context`. Un

contesto di crittografia è un set di coppie chiave-valore che possono contenere ulteriori informazioni contestuali sui dati.

Amazon S3 utilizza automaticamente il nome della risorsa Amazon (ARN) dell'oggetto come coppia di contesto di crittografia; ad esempio, `arn:aws:s3:::object_ARN`.

Facoltativamente, è possibile fornire una coppia di contesto di crittografia aggiuntiva utilizzando l'intestazione `x-amz-server-side-encryption-context`. Tuttavia, poiché il contesto di crittografia non è crittografato, assicurati che non includa informazioni sensibili. Amazon S3 archivia questa coppia di chiavi aggiuntiva insieme al contesto di crittografia predefinito.

Per informazioni sul contesto di crittografia in Amazon S3, consulta la sezione Contesto di crittografia. Per informazioni generali sul contesto di crittografia, consulta Concetti di AWS Key Management Service : Contesto di crittografia nella Guida per gli sviluppatori di AWS Key Management Service .

AWS KMS ID chiave () **`x-amz-server-side-encryption-aws-kms-key-id`**

Puoi utilizzare l'intestazione `x-amz-server-side-encryption-aws-kms-key-id` per specificare l'ID della chiave gestita dal cliente utilizzata per proteggere i dati. Se specifichi l'`x-amz-server-side-encryption:aws:kms:dsse`intestazione ma non la `x-amz-server-side-encryption-aws-kms-key-id` fornisci, Amazon S3 utilizza `aws/s3` () per Chiave gestita da AWS proteggere i dati. Se desideri utilizzare una chiave gestita dal cliente, devi fornire l'intestazione `x-amz-server-side-encryption-aws-kms-key-id` della chiave gestita dal cliente.

> **Important**
>
> Quando utilizzi una chiave KMS AWS KMS key per la crittografia lato server in Amazon S3, devi scegliere una chiave KMS di crittografia simmetrica. Amazon S3 supporta solo chiavi KMS di crittografia simmetrica. Per ulteriori informazioni sulle chiavi, consulta Chiavi KMS di crittografia simmetrica nella Guida per gli sviluppatori di AWS Key Management Service .

Usando il AWS CLI

Quando carichi un nuovo oggetto o copi un oggetto esistente, puoi specificare l'uso di DSSE-KMS per crittografare i dati. Per farlo, aggiungi il parametro `--server-side-encryption aws:kms:dsse` alla richiesta. Usa il parametro `--ssekms-key-id` *`example-key-id`* per aggiungere la chiave AWS KMS gestita dal cliente che hai creato. Se specifichi `--server-side-encryption aws:kms:dsse` ma non fornisci un ID di AWS KMS chiave, Amazon S3 utilizzerà la chiave AWS gestita ()`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
```

È possibile crittografare un oggetto non crittografato con DSSE-KMS copiando nuovamente l'oggetto nella sua posizione.

```
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
```

Utilizzo della crittografia lato server con chiavi fornite dal cliente (SSE-C)

La crittografia lato server consente di proteggere i dati inattivi. La crittografia lato server viene applicata solo ai dati dell'oggetto, non dei metadati dell'oggetto. Utilizzando la crittografia lato server con chiavi fornite dal cliente (SSE-C), puoi archiviare i dati crittografati con le tue chiavi di crittografia. Con la chiave di crittografia fornita come parte della richiesta, Amazon S3 gestisce la crittografia dei dati durante le operazioni di scrittura su disco e decrittografia dei dati quando viene eseguito l'accesso agli oggetti. Pertanto, non è necessario mantenere alcun codice per effettuare la crittografia e la decrittografia dei dati. L'unica cosa che rimane da fare è gestire le chiavi di crittografia fornite.

Quando viene caricato un oggetto, Amazon S3 utilizza la chiave di crittografia fornita per applicare la crittografia AES-256 ai dati. Amazon S3 rimuove quindi la chiave di crittografia dalla memoria. Quando viene recuperato un oggetto, è necessario fornire la stessa chiave di crittografia come parte della richiesta. Amazon S3 verifica prima che la chiave di crittografia fornita corrisponda, quindi esegue la decrittografia dell'oggetto prima di restituire i relativi dati.

Questa caratteristica non comporta costi supplementari per l'utilizzo di SSE-C. Tuttavia, le richieste di configurazione e utilizzo di SSE-C sono soggette alle tariffe standard delle richieste Amazon S3. Per informazioni sui prezzi, consulta Prezzi di Amazon S3.

> **Note**
>
> Amazon S3 non archivia le chiavi di crittografia fornite. Archivia invece un valore per il codice di autenticazione dei messaggi basato su hash (HMAC) con salting casuale della chiave di crittografia per convalidare le richieste future. Il valore HMAC con l'introduzione di un sale non può essere utilizzato per derivare il valore della chiave di crittografia o per decrittografare i contenuti dell'oggetto crittografato. Ciò significa che se si perde la chiave di crittografia, si perde l'oggetto.

S3 Replication supporta gli oggetti crittografati con SSE-C. Per ulteriori informazioni sulla replica di oggetti crittografati, consulta the section called “Replica di oggetti crittografati”.

Per ulteriori informazioni su SSE-C, consulta i seguenti argomenti.

Argomenti

- Panoramica di SSE-C
- Richiesta e limitazione di SSE-C
- URL prefirmati e SSE-C
- Specifica della crittografia lato server con chiavi fornite dal cliente (SSE-C)

Panoramica di SSE-C

In questa sezione viene fornita una panoramica di SSE-C. Quando si utilizza SSE-C, è necessario tenere presente le considerazioni riportate di seguito.

- È necessario utilizzare HTTPS.

> **Important**
>
> Amazon S3 rifiuta qualsiasi richiesta effettuata su HTTP quando si utilizza SSE-C. Per motivi di sicurezza, è consigliabile considerare compromessa qualsiasi chiave inviata per errore tramite HTTP. Elimina la chiave ed esegui la rotazione come opportuno.

- Il tag di entità (ETag) nella risposta non è l'hash MD5 dei dati degli oggetti.
- L'utente gestisce una mappatura per tenere traccia della chiave di crittografia che è stata utilizzata per crittografare un determinato oggetto. Amazon S3 non archivia le chiavi di crittografia. L'utente è responsabile della tracciatura di ciascuna chiave di crittografia fornita per ogni determinato oggetto.
  - Se per il bucket in uso è abilitata la funzione di controllo delle versioni, ogni versione di oggetto caricata utilizzando questa caratteristica può avere la propria chiave di crittografia. L'utente è responsabile della tracciatura di ciascuna chiave di crittografia utilizzata per ogni determinato oggetto.
  - Dato che l'utente gestisce le chiavi di crittografia lato cliente, gestisce anche eventuali tutele aggiuntive, come la rotazione delle chiavi, lato cliente.

> **Warning**
>
> Se la chiave di crittografia viene smarrita, qualsiasi richiesta GET di un determinato oggetto senza la rispettiva chiave di crittografia non va a buon fine e l'oggetto viene perduto.

Richiesta e limitazione di SSE-C

Per richiedere le chiavi SSE-C per tutti gli oggetti in uno specifico bucket Amazon S3, è possibile utilizzare una policy di bucket.

Ad esempio, la seguente policy del bucket rifiuta l'autorizzazione per il caricamento di oggetti (`s3:PutObject`) per tutte le richieste che non includono l'intestazione `x-amz-server-side-encryption-customer-algorithm` che richiede di SSE-C.

```
{
    "Version": "2012-10-17",
    "Id": "PutObjectPolicy",
    "Statement": [
        {
            "Sid": "RequireSSECObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::example-s3-bucket/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption-customer-algorithm": "true"
                }
            }
        }
    ]
}
```

Per limitare la crittografia lato server di tutti gli oggetti in uno specifico bucket Amazon S3, è anche possibile utilizzare una policy. Ad esempio, la seguente policy di bucket rifiuta a chiunque l'autorizzazione al caricamento dell'oggetto (`s3:PutObject`) se la richiesta non include l'intestazione `x-amz-server-side-encryption-customer-algorithm` che richiede la crittografia con chiavi SSE-C.

```
{
    "Version": "2012-10-17",
    "Id": "PutObjectPolicy",
    "Statement": [
        {
            "Sid": "RestrictSSECObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::example-s3-bucket/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption-customer-algorithm": "false"
                }
            }
        }
    ]
}
```

Important

Se utilizzi una policy bucket per richiedere l'attivazione di SSE-C`s3:PutObject`, devi includere l'`x-amz-server-side-encryption-customer-algorithm`intestazione in tutte le richieste di caricamento in più parti (, e). CreateMultipartUpload UploadPart CompleteMultipartUpload

URL prefirmati e SSE-C

È possibile generare un URL prefirmato che possa essere utilizzato per operazioni quali il caricamento di un nuovo oggetto, il recupero di un oggetto esistente o dei metadata di un oggetto. Gli URL prefirmati supportano le chiavi SSE-C come segue:

- Quando viene creato un URL prefirmato, è necessario specificare l'algoritmo utilizzando l'intestazione `x-amz-server-side-encryption-customer-algorithm` nel calcolo della firma.
- Quando viene utilizzato l'URL prefirmato per il caricamento di un nuovo oggetto, il recupero di un oggetto esistente o solo dei metadata di un oggetto, è necessario fornire tutte le intestazioni di crittografia nella richiesta dell'applicazione client.

> **Note**
>
> Per gli oggetti non SSE-C, puoi generare un URL prefirmato e copiarlo direttamente nel browser per accedere ai dati.
> Tuttavia, ciò non è possibile per gli oggetti SSE-C poiché oltre all'URL prefirmato, è anche necessario includere le intestazioni HTTP specifiche degli oggetti SSE-C. È quindi possibile utilizzare URL prefirmati per gli oggetti SSE-C solo a livello di programmazione.

Per ulteriori informazioni sugli URL prefirmati, consulta the section called “Utilizzo di URL prefirmati”.

Specifica della crittografia lato server con chiavi fornite dal cliente (SSE-C)

Al momento della creazione di oggetti con REST API, è possibile specificare la crittografia lato server con le chiavi fornite dal cliente (SSE-C). Quando si utilizza SSE-C, è necessario fornire le informazioni sulla chiave di crittografia utilizzando le intestazioni di richiesta seguenti.

Nome	Descrizione
`x-amz-server-side-encryption-customer-algorithm`	Utilizzare questa intestazione per specificare l'algoritmo di crittografia. Il valore dell'intestazione deve essere `AES256`.
`x-amz-server-side-encryption-customer-key`	Utilizzare questa intestazione per fornire la chiave di crittografia a 256 bit codificata con base64 per consentire ad Amazon S3 di crittografare o decrittare i dati.
`x-amz-server-side-encryption-customer-key-MD5`	Utilizzare questa intestazione per fornire il digest MD5 a 128 bit codificato con base64 della chiave di crittografia secondo RFC 1321. Amazon S3 utilizza questa intestazione per il controllo dell'integrità del messaggio per accertarsi che la chiave di crittografia sia stata trasmessa senza errori.

Puoi usare le librerie wrapper AWS SDK per aggiungere queste intestazioni alla tua richiesta.
Se necessario, è possibile anche effettuare le chiamate REST API di Amazon S3 direttamente nell'applicazione.

> **Note**
>
> Non è inoltre possibile utilizzare la console di Amazon S3 per aggiornare (ad esempio, cambiare la classe di archiviazione o aggiungere metadati) un oggetto archiviato esistente utilizzando la crittografia SSE-C.

Utilizzo di REST API

REST API di Amazon S3 che supportano SSE-C

Le seguenti API Amazon S3 supportano la crittografia lato server con le chiavi di crittografia fornite dal cliente (SSE-C).

- Operazione GET: quando si recuperano oggetti utilizzando l'API GET (consulta l'argomento relativo all'operazione GetObject), è possibile specificare le intestazioni di richiesta.
- Operazione HEAD: per recuperare i metadati dell'oggetto utilizzando l'API HEAD (consulta l'argomento relativo all'operazione HeadObject), è possibile specificare queste intestazioni di richiesta.
- Operazione PUT: quando si caricano dati utilizzando l'API PUT (consulta l'argomento relativo all'operazione PutObject), è possibile specificare queste intestazioni di richiesta.
- Caricamento in più parti: quando si caricano oggetti di grandi dimensioni utilizzando l'API per il caricamento in più parti, è possibile specificare queste intestazioni. È necessario specificare queste intestazioni nella richiesta di avvio (consulta l'argomento relativo all'avvio di caricamenti in più parti) e in ogni richiesta di caricamento di parti successive (consulta l'argomento relativo al caricamento delle parti o caricamento delle copie di parti). Per ogni richiesta di caricamento di parte, le informazioni della crittografia devono essere uguali a quelle specificate nella richiesta di avvio di caricamento in più parti.
- Operazione POST: quando si utilizza un'operazione POST per caricare un oggetto (consulta l'argomento relativo all'oggetto POST), anziché nelle intestazioni di richiesta, è necessario specificare le stesse informazioni nei campi del modulo.
- Operazione COPY: l'operazione di copia di un oggetto (consulta l'argomento relativo all'operazione CopyObject) interessa un oggetto di origine e un oggetto di destinazione.
  - Se desideri che l'oggetto di destinazione sia crittografato utilizzando la crittografia lato server con chiavi AWS gestite, devi fornire l'intestazione della richiesta. `x-amz-server-side-encryption`

- Se si vuole crittografare l'oggetto di destinazione utilizzando SSE-C, è necessario fornire le informazioni della crittografia utilizzando le tre intestazioni descritte nella tabella precedente.
- Se l'oggetto di origine è crittografato con SSE-C, è necessario fornire le informazioni relative alle chiavi di crittografia utilizzando le seguenti intestazioni affinché Amazon S3 possa decrittare l'oggetto per copiarlo.

Nome	Descrizione
`x-amz-copy-source-server-side-encryption-customer-algorithm`	Includere questa intestazione per specificare l'algoritmo che dovrebbe utilizzare Amazon S3 per decrittare l'oggetto di origine. Questo valore deve essere `AES256`.
`x-amz-copy-source-server-side-encryption-customer-key`	Includere questa intestazione per fornire la chiave di crittografia codificata con base64 per consentire ad Amazon S3 di decrittare l'oggetto di origine. Questa chiave di crittografia deve essere quella fornita ad Amazon S3 quando è stato creato l'oggetto di origine. In caso contrario, Amazon S3 non riesce a decrittare l'oggetto.
`x-amz-copy-source-server-side-encryption-customer-key-MD5`	Includere questa intestazione per fornire il digest MD5 a 128 bit codificato con base64 della chiave di crittografia secondo RFC 1321.

Utilizzo degli AWS SDK per specificare SSE-C per le operazioni PUT, GET, Head e Copy

Nel seguente esempio viene mostrato come richiedere la crittografia lato server con chiavi fornite dal cliente (SSE-C) per gli oggetti. Negli esempi vengono eseguite le operazioni riportate di seguito. Per ogni operazione viene illustrato come specificare le intestazioni correlate alle chiavi SSE-C nella richiesta:

- Put object: consente di caricare un oggetto e richiedere la crittografia lato server utilizzando la chiave di crittografia fornita dal cliente.

- Get object: consente di scaricare l'oggetto caricato durante la fase precedente. Nella richiesta, vengono fornite le stesse informazioni di crittografia specificate quando è stato caricato l'oggetto. per consentire ad Amazon S3 di decrittare l'oggetto e di restituirlo.
- Get object metadata: consente di recuperare i metadati dell'oggetto. Fornire le stesse informazioni di crittografia utilizzate quando l'oggetto è stato creato.
- Copy object: consente di creare una copia dell'oggetto caricato in precedenza. Poiché l'oggetto di origine è stato archiviato utilizzando la chiave SSE-C, è necessario fornire le relative informazioni di crittografia nella richiesta di copia. Per impostazione predefinita, Amazon S3 esegue la crittografia dell'oggetto solo se richiesta esplicitamente. In questo esempio, Amazon S3 viene configurato per archiviare una copia crittografata dell'oggetto.

Java

> **Note**
>
> In questo esempio viene illustrato come caricare un oggetto in un'unica operazione. Quando si utilizza l'API per il caricamento in più parti per caricare oggetti di grandi dimensioni, fornire le informazioni di crittografia nello stesso modo mostrato in questo esempio. Per esempi di caricamenti in più parti che utilizzano il, consulta. AWS SDK for JavaCaricamento di un oggetto utilizzando il caricamento in più parti

Per aggiungere le informazioni di crittografia richieste, includere `SSECustomerKey` nella richiesta. Per ulteriori informazioni sulla classe `SSECustomerKey`, consulta la sezione relativa a REST API.

Per informazioni su SSE-C, consulta Utilizzo della crittografia lato server con chiavi fornite dal cliente (SSE-C). Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

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**
>
> Per esempi di caricamento di oggetti di grandi dimensioni utilizzando l'API per il caricamento in più parti, consulta Caricamento di un oggetto utilizzando il caricamento in più parti e Utilizzo degli AWS SDK (API di basso livello).

Per informazioni su SSE-C, consulta Utilizzo della crittografia lato server con chiavi fornite dal cliente (SSE-C). Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

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);
        }
    }
}
```

Utilizzo degli AWS SDK per specificare SSE-C per caricamenti multiparte

L'esempio nella sezione precedente mostra come richiedere la crittografia lato server con la chiave fornita dal cliente (SSE-C) nelle operazioni PUT, GET, Head e Copy. In questa sezione vengono descritte altre API Amazon S3 che supportano la chiave SSE-C.

Java

Per caricare oggetti di grandi dimensioni, è possibile utilizzare l'API per il caricamento in più parti (consulta Caricamento e copia di oggetti utilizzando il caricamento in più parti). È possibile

utilizzare indifferentemente API di livello alto o basso per caricare gli oggetti di grandi dimensioni. Queste API supportano l'uso di intestazioni correlate alla crittografia nella richiesta.

- Quando si utilizza l'API `TransferManager` di alto livello, è necessario fornire le intestazioni specifiche della crittografia in `PutObjectRequest` (consulta Caricamento di un oggetto utilizzando il caricamento in più parti).
- Quando si utilizza l'API di basso livello, è necessario fornire informazioni correlate alla crittografia nella `InitiateMultipartUploadRequest`, seguite da informazioni di crittografia identiche in ogni `UploadPartRequest`. Non è necessario fornire alcuna intestazione specifica della crittografia nella `CompleteMultipartUploadRequest`. Per alcuni esempi, consulta Utilizzo degli AWS SDK (API di basso livello).

Nel seguente esempio viene utilizzato `TransferManager` per creare gli oggetti e viene illustrato come fornire le informazioni correlate alla chiave SSE-C. Inoltre, vengono effettuate le seguenti operazioni:

- Viene creato un oggetto utilizzando il metodo `TransferManager.upload()`. Nell'istanza `PutObjectRequest`, fornire le informazioni sulla chiave di crittografia da richiedere. Amazon S3 esegue la crittografia dell'oggetto utilizzando la chiave fornita dal cliente.
- Viene eseguita una copia dell'oggetto richiamando il metodo `TransferManager.copy()`. Nell'esempio Amazon S3 viene configurato per crittografare la copia dell'oggetto utilizzando una nuova `SSECustomerKey`. Poiché l'oggetto di origine è crittografato tramite la chiave SSE-C, `CopyObjectRequest` fornisce anche la chiave di crittografia dell'oggetto di origine in modo che Amazon S3 possa decrittare l'oggetto prima di 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

Per caricare oggetti di grandi dimensioni, puoi utilizzare l'API di caricamento multiparte (vedi). Caricamento e copia di oggetti utilizzando il caricamento in più parti AWS SDK for .NET fornisce API di alto o basso livello per caricare oggetti di grandi dimensioni. Queste API supportano l'uso di intestazioni correlate alla crittografia nella richiesta.

- Quando si utilizza l'API `Transfer-Utility` di alto livello, è necessario fornire le intestazioni specifiche della crittografia in `TransferUtilityUploadRequest` come mostrato. Per alcuni esempi di codice, consulta Caricamento di un oggetto utilizzando il caricamento in più parti.

```
TransferUtilityUploadRequest request = new TransferUtilityUploadRequest()
{
    FilePath = filePath,
    BucketName = existingBucketName,
    Key = keyName,
    // Provide encryption information.
    ServerSideEncryptionCustomerMethod =
 ServerSideEncryptionCustomerMethod.AES256,
    ServerSideEncryptionCustomerProvidedKey = base64Key,
};
```

- Quando si utilizza l'API di basso livello, è necessario fornire informazioni correlate alla crittografia nella richiesta di avvio del caricamento in più parti, seguite da informazioni di crittografia identiche nelle successive richieste di caricamento della parte. Non è necessario fornire alcuna intestazione specifica della crittografia nella richiesta di caricamento in più parti completa. Per alcuni esempi, consulta Utilizzo degli AWS SDK (API di basso livello).

  Di seguito è riportato un esempio di caricamento in più parti di basso livello in cui viene creata una copia di un oggetto di grandi dimensioni esistente. Nell'esempio l'oggetto da copiare è archiviato in Amazon S3 con la chiave SSE-C e l'oggetto di destinazione deve essere salvato utilizzando la stessa chiave. Nell'esempio vengono effettuate le seguenti operazioni:

  - Viene avviata una richiesta di caricamento in più parti specificando una chiave di crittografia e informazioni correlate.
  - Vengono fornite chiavi di crittografia e informazioni correlate per gli oggetti di origine e di destinazione nella richiesta `CopyPartRequest`.
  - Viene ottenuta la dimensione dell'oggetto di origine da copiare recuperando i metadata dell'oggetto.
  - Caricamento degli oggetti in parti da 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);
            }
        }
    }
}
```

Protezione dei dati con la crittografia lato client

La crittografia lato client è l'atto di crittografare i dati a livello locale per garantirne la sicurezza in transito e a riposo. Per crittografare gli oggetti prima di inviarli ad Amazon S3, utilizza il client di crittografia Amazon S3. Quando gli oggetti vengono crittografati in questo modo, non vengono esposti a terzi, inclusi. AWS Amazon S3 riceve i tuoi oggetti già crittografati e non ha un ruolo nella crittografia o decrittografia degli oggetti. Puoi utilizzare sia il client di crittografia Amazon S3 che la crittografia lato server per crittografare i tuoi dati. Quando invii oggetti crittografati ad Amazon S3, non vengono riconosciuti come crittografati e vengono rilevati solo gli oggetti tipici.

Il client di crittografia Amazon S3 funge da intermediario tra te e Amazon S3. Dopo aver creato l'istanza del client di crittografia Amazon S3, i tuoi oggetti vengono automaticamente crittografati e decrittati come parte delle richieste `PutObject` e `GetObject` di Amazon S3. I tuoi oggetti sono tutti crittografati con una chiave di dati univoca. Il client di crittografia Amazon S3 non utilizza né interagisce con le chiavi bucket, anche se si specifica una chiave KMS come chiave di wrapping.

La Guida per gli sviluppatori del client di crittografia Amazon S3 si concentra sulle versioni 3.0 e successive del client di crittografia Amazon S3. Per ulteriori informazioni, consulta Cos'è il client di crittografia Amazon S3 nella Guida per gli sviluppatori di Client di crittografia Amazon S3.

Per ulteriori informazioni sulle versioni precedenti del client di crittografia Amazon S3, consulta la AWS SDK Developer Guide per il tuo linguaggio di programmazione.

- AWS SDK for Java
- AWS SDK for .NET
- AWS SDK for Go
- AWS SDK for PHP
- AWS SDK for Ruby
- AWS SDK for C++

Riservatezza del traffico Internet

Questo argomento descrive come Amazon S3 protegge le connessioni dal servizio ad altri percorsi.

Traffico tra servizio e applicazioni e client locali

È possibile combinare le seguenti connessioni AWS PrivateLink per fornire connettività tra la rete privata e: AWS

- Una AWS connessione VPN da sito a sito. Per ulteriori informazioni, vedi Cos'è? AWS Site-to-Site VPN
- Una AWS Direct Connect connessione. Per ulteriori informazioni, vedi Cos'è AWS Direct Connect?

L'accesso ad Amazon S3 tramite la rete avviene tramite API AWS pubblicate. I client devono supportare Transport Layer Security (TLS) 1.2. È consigliabile TLS 1.3. I client devono inoltre supportare le suite di cifratura con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). La maggior parte dei sistemi moderni come Java 7 e versioni successive, supporta tali modalità. Inoltre, è necessario firmare le richieste utilizzando un ID chiave di accesso e la chiave di accesso segreta associate a un principale IAM, oppure è possibile utilizzare AWS Security Token Service (STS) per generare le credenziali di sicurezza temporanee per firmare le richieste.

Traffico tra AWS risorse nella stessa regione

Un endpoint Virtual Private Cloud (VPC) Amazon S3 è un'entità logica all'interno di un VPC che consente la connettività solo ad Amazon S3 Il VPC instrada le richieste ad Amazon S3 e le risposte al VPC. Per ulteriori informazioni, consulta Endpoint VPC nella Guida per l'utente di VPC. Per policy del bucket di esempio che puoi utilizzare per controllare l'accesso ai bucket S3 da endpoint VPC, consulta Controllo dell'accesso dagli endpoint VPC con policy di bucket.

AWS PrivateLink per Amazon S3

Con AWS PrivateLink Amazon S3, puoi fornire endpoint VPC di interfaccia (endpoint di interfaccia) nel tuo cloud privato virtuale (VPC). Questi endpoint sono accessibili direttamente dalle applicazioni locali tramite VPN e/o in un altro modo di peering Regione AWS tramite VPC. AWS Direct Connect

Gli endpoint di interfaccia sono rappresentati da una o più interfacce di rete elastiche (ENI) a cui vengono assegnati indirizzi IP privati dalle sottoreti nel VPC. Le richieste ad Amazon S3 tramite gli

endpoint di interfaccia rimangono nella rete Amazon. Puoi anche accedere agli endpoint di interfaccia nel tuo VPC da applicazioni locali AWS Direct Connect tramite AWS Virtual Private Network o ().AWS VPN Per ulteriori informazioni su come connettere il VPC alla rete On-Premise, consulta la AWS Direct Connect Guida per l'utente di e la AWS Site-to-Site VPN Guida per l'utente di .

Per informazioni sulla creazione di endpoint di interfaccia, consulta Endpoint VPC di interfaccia (AWS PrivateLink) nella Guida di AWS PrivateLink .

Argomenti

- Tipi di endpoint VPC per Amazon S3
- Restrizioni e limitazioni di AWS PrivateLink per Amazon S3
- Creazione di un endpoint VPC
- Accesso agli endpoint di interfaccia di Amazon S3
- DNS privato
- Accesso ai bucket, ai punti di accesso e alle operazioni API di controllo Amazon S3 dagli endpoint di interfaccia S3
- Aggiornamento di una configurazione DNS locale
- Creazione di una policy per l'endpoint VPC per Amazon S3

Tipi di endpoint VPC per Amazon S3

Puoi utilizzare due tipi di endpoint VPC per accedere ad Amazon S3: endpoint gateway ed endpoint di interfaccia (utilizzando). AWS PrivateLink Un endpoint gateway è un gateway specificato nella tabella di routing per accedere ad Amazon S3 dal tuo VPC tramite la rete. AWS Gli endpoint di interfaccia estendono la funzionalità degli endpoint gateway utilizzando indirizzi IP privati per instradare le richieste ad Amazon S3 dall'interno del tuo VPC, in locale o da un VPC in un altro tramite peering VPC o. Regione AWS AWS Transit Gateway Per ulteriori informazioni, consulta Che cos'è il peering di VPC? e Transit Gateway e peering di VPC.

Gli endpoint di interfaccia sono compatibili con gli endpoint gateway. Se disponi di un endpoint gateway nel VPC, puoi utilizzare entrambi i tipi di endpoint nello stesso VPC.

Endpoint gateway per Amazon S3	Endpoint di interfaccia per Amazon S3
In entrambi i casi, il traffico di rete rimane sulla rete. AWS	

Endpoint gateway per Amazon S3	Endpoint di interfaccia per Amazon S3
Uso di indirizzi IP pubblici di Amazon S3	Uso di indirizzi IP privati del tuo VPC per accedere ad Amazon S3
Uso degli stessi nomi DNS di Simple Storage Service (Amazon S3)	Richiesta di nomi DNS di Simple Storage Service (Amazon S3) specifici per endpoint
Non consente l'accesso da on-premise	Consente l'accesso da On-Premise
Non consentire l'accesso da parte di un altro Regione AWS	Consenti l'accesso da un VPC a un altro Regione AWS utilizzando il peering VPC o AWS Transit Gateway
Non fatturata	Fatturata

Per ulteriori informazioni sugli endpoint gateway, consulta Endpoint VPC gateway nella Guida di AWS PrivateLink .

Restrizioni e limitazioni di AWS PrivateLink per Amazon S3

Le limitazioni VPC si applicano AWS PrivateLink ad Amazon S3. Per ulteriori informazioni, consulta Considerazioni di un endpoint di interfaccia e Quote di AWS PrivateLink nella Guida di AWS PrivateLink . Inoltre, si applicano le limitazioni seguenti:

AWS PrivateLink per Amazon S3 non supporta quanto segue:

- Endpoint FIPS (Federal Information Processing Standard)
- Endpoint del sito Web
- Endpoint globali legacy
- Endpoint di regione S3 Dash
- Endpoint dual-stack Amazon S3
- Utilizzo CopyObjecto UploadPartCopytra bucket in diversi Regioni AWS
- Transport Layer Security (TLS) 1.1

Creazione di un endpoint VPC

Per creare un endpoint di interfaccia VPC, consulta Creazione di un endpoint VPC nella Guida AWS PrivateLink .

Accesso agli endpoint di interfaccia di Amazon S3

Quando crei un endpoint di interfaccia, Amazon S3 genera due tipi di nomi DNS S3 specifici dell'endpoint: regionale e zonale.

- Un nome DNS regionale include un ID endpoint VPC univoco, un identificatore di servizio, Regione AWS`vpce.amazonaws.com` il e nel nome. Ad esempio, per l'ID endpoint VPC *`vpce-1a2b3c4d`*, il nome DNS generato potrebbe essere simile a *`vpce-1a2b3c4d-5e6f`*`.s3.us-east-1.vpce.amazonaws.com`.
- I nomi DNS zonali includono la zona di disponibilità, ad esempio *`vpce-1a2b3c4d-5e6f`***`-us-east-1a.`**`s3.us-east-1.vpce.amazonaws.com`. Puoi utilizzare questa opzione se l'architettura isola le zone di disponibilità. Ad esempio, puoi utilizzarla per il contenimento degli errori o per ridurre i costi di trasferimento dei dati a livello regionale.

I nomi DNS S3 specifici degli endpoint possono essere risolti dal dominio DNS pubblico S3.

DNS privato

Le opzioni DNS private per gli endpoint di interfaccia VPC semplificano l'instradamento del traffico S3 sugli endpoint VPC e consentono di sfruttare il percorso di rete più economico disponibile per l'applicazione. Puoi utilizzare le opzioni DNS private per indirizzare il traffico regionale S3 senza aggiornare i client S3 per usare i nomi DNS specifici degli endpoint di interfaccia o gestire l'infrastruttura DNS. Con i nomi DNS privati abilitati, le query DNS regionali di S3 vengono risolte negli indirizzi IP privati dei seguenti endpoint: AWS PrivateLink

- Endpoint di bucket regionale (ad esempio, `s3.us-east-1.amazonaws.com`)
- Endpoint di controllo (ad esempio, `s3-control.us-east-1.amazonaws.com`)
- Endpoint di punto di accesso (ad esempio, `s3-accesspoint.us-east-1.amazonaws.com`)

Se hai un endpoint gateway nel tuo VPC, puoi indirizzare automaticamente le richieste in entrata nel VPC all'endpoint gateway S3 esistente e le richieste on-premise all'endpoint di interfaccia. Questo approccio consente di ottimizzare i costi di rete utilizzando gli endpoint gateway, che non vengono

fatturati, per il traffico in entrata nel VPC. Le applicazioni locali possono essere utilizzate AWS PrivateLink con l'aiuto dell'endpoint Resolver in entrata. Amazon fornisce un server DNS chiamato il Route 53 Resolver per il tuo VPC. Un endpoint del resolver in entrata inoltra le query DNS dalla rete on-premise al Route 53 Resolver.

> **Important**
>
> Per sfruttare il percorso di rete più economico quando si utilizza Abilita DNS privato solo per gli endpoint in entrata, è necessario che nel cloud privato virtuale sia presente un endpoint gateway. La presenza di un endpoint gateway aiuta a garantire che il traffico in entrata nel VPC venga sempre indirizzato sulla rete privata AWS quando è selezionata l'opzione Abilita DNS privato solo per gli endpoint in entrata. È necessario mantenere questo endpoint gateway se è selezionata l'opzione Abilita DNS privato solo per gli endpoint in entrata. Se desideri eliminare l'endpoint gateway, devi prima deselezionare Abilita DNS privato solo per gli endpoint in entrata.
> Se desideri aggiornare un endpoint di interfaccia esistente su Abilita DNS privato solo per gli endpoint in entrata verifica innanzitutto che il tuo VPC disponga di un endpoint gateway S3. Per ulteriori informazioni sugli endpoint gateway e sulla gestione dei nomi DNS privati, consulta rispettivamente Endpoint gateway del VPC e Gestione dei nomi DNS nella Guida di AWS PrivateLink .

L'opzione Abilita DNS privato solo per gli endpoint in entrata è disponibile solo per i servizi che supportano gli endpoint gateway.

Per ulteriori informazioni sulla creazione di un endpoint VPC che utilizza Abilita DNS privato solo per gli endpoint in entrata, consulta Creare un endpoint di interfaccia nella Guida di AWS PrivateLink .

Utilizzo della console VPC

Nella console sono disponibili due opzioni: Abilita nome DNS e Abilita DNS privato solo per gli endpoint in entrata. Abilita il nome DNS è un'opzione supportata da. AWS PrivateLink Con l'opzione Abilita nome DNS puoi utilizzare la connettività privata di Amazon ad Amazon S3, effettuando richieste ai nomi DNS predefiniti degli endpoint pubblici. Quando questa opzione è abilitata, i clienti possono sfruttare il percorso di rete più economico disponibile per la loro applicazione.

Quando abiliti i nomi DNS privati su un endpoint di interfaccia VPC esistente o nuovo per Amazon S3, l'opzione Abilita DNS privato solo per gli endpoint in entrata è selezionata per impostazione

predefinita. Se questa opzione è selezionata, le applicazioni utilizzano solo gli endpoint di interfaccia per il traffico on-premise. Il traffico VPC in entrata utilizza automaticamente gli endpoint gateway più economici. In alternativa, puoi deselezionare Abilita DNS privato solo per gli endpoint in entrata per indirizzare tutte le richieste S3 sull'endpoint di interfaccia.

Usando il AWS CLI

Se non specifichi un valore per `PrivateDnsOnlyForInboundResolverEndpoint`, viene usata l'impostazione predefinita `true`. Tuttavia, prima che il cloud privato virtuale applichi le impostazioni, esegue un controllo per assicurarsi che nel cloud privato VPC sia presente un endpoint gateway. Se nel cloud privato virtuale è presente un endpoint gateway, la chiamata ha esito positivo. In caso contrario, viene visualizzato il seguente messaggio di errore:

Per essere impostato su PrivateDnsOnlyForInboundResolverEndpoint true, il VPC *`vpce_id`* deve disporre di un endpoint gateway per il servizio.

Per un nuovo endpoint di interfaccia VPC

Usa gli attributi `private-dns-enabled` e `dns-options` per abilitare il DNS privato tramite la linea di comando. L'opzione `PrivateDnsOnlyForInboundResolverEndpoint` nell'attributo `dns-options` deve essere impostata su `true`. Sostituire *`user input placeholders`* con le proprie informazioni.

```
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
```

Per un endpoint VPC esistente

Se desideri utilizzare il DNS privato per un endpoint VPC esistente, usa il seguente comando di esempio e sostituisci *`user input placeholders`* con le tue specifiche informazioni.

```
aws ec2 modify-vpc-endpoint \
--region us-east-1 \
--vpc-endpoint-id client-vpc-id \
--private-dns-enabled \
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=false
```

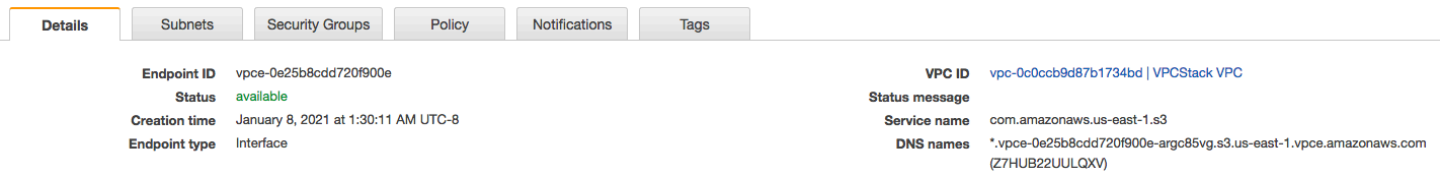
Se desideri aggiornare un endpoint VPC esistente per abilitare il DNS privato solo per il risolutore in entrata, usa il seguente esempio e sostituisci i valori di esempio con le tue specifiche informazioni.

```
aws ec2 modify-vpc-endpoint \
--region us-east-1 \
--vpc-endpoint-id client-vpc-id \
--private-dns-enabled \
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=true
```

Accesso ai bucket, ai punti di accesso e alle operazioni API di controllo Amazon S3 dagli endpoint di interfaccia S3

Puoi utilizzare i nostri AWS SDK AWS CLI per accedere a bucket, punti di accesso S3 e operazioni dell'API Amazon S3 Control tramite gli endpoint dell'interfaccia S3.

Nell'immagine seguente viene illustrata la scheda Dettagli della console VPC, in cui è possibile trovare il nome DNS di un endpoint VPC. In questo esempio, l'ID endpoint VPC (vpce-id) è `vpce-0e25b8cdd720f900e` e il nome DNS è `*.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com`.



Quando utilizzi il nome DNS per accedere a una risorsa, sostituisci * con il valore appropriato. I valori appropriati da utilizzare al posto di * sono i seguenti:

- `bucket`
- `accesspoint`
- `control`

Ad esempio, per accedere a un bucket, usa un nome DNS simile al seguente:

`bucket.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com`

Per esempi di come utilizzare i nomi DNS per accedere a bucket, punti di accesso e operazioni API di controllo Amazon S3, consulta le sezioni AWS CLI esempi e AWS Esempi SDK.

Per ulteriori informazioni su come visualizzare i nomi DNS specifici degli endpoint, consulta Visualizzazione della configurazione dei nomi DNS privati del servizio endpoint nella Guida per l'utente di VPC.

AWS CLI esempi

Per accedere ai bucket S3, ai punti di accesso S3 o alle operazioni dell'API Amazon S3 Control tramite gli endpoint dell'interfaccia S3 nei AWS CLI comandi, utilizza i parametri and. `--region` `--endpoint-url`

Esempio: utilizzo dell'URL dell'endpoint per elencare gli oggetti nel bucket

Nell'esempio seguente, sostituisci il nome bucket *`my-bucket`*, Regione *`us-east-1`* e il nome DNS dell'ID endpoint VPC *`vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com`* con le tue informazioni.

```
aws s3 ls s3://my-bucket/ --region us-east-1 --endpoint-url
 https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com
```

Esempio: utilizzo dell'URL dell'endpoint per elencare gli oggetti da un punto di accesso

- Metodo 1: utilizzo del nome della risorsa Amazon (ARN) del punto di accesso con l'endpoint del punto di accesso

  Sostituisci l'ARN *`us-east-1:123456789012:accesspoint/accesspointexamplename`*, la Regione *`us-east-1`* e l'ID endpoint VPC *`vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com`* con le informazioni appropriate.

```
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
```

Se non riesci a eseguire correttamente il comando, aggiorna il comando AWS CLI alla versione più recente e riprova. Per ulteriori informazioni sulle istruzioni di aggiornamento, consulta Istruzioni per l'installazione o l'aggiornamento all'ultima versione della AWS CLI nella Guida per l'utente di AWS Command Line Interface .

- Metodo 2: utilizzo dell'alias del punto di accesso con l'endpoint bucket regionale

  Nell'esempio seguente, sostituisci l'alias del punto di accesso *accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias*, la Regione *us-east-1* e l'ID endpoint VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con le informazioni appropriate.

  ```
  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
  ```

- Metodo 3: utilizzo dell'alias del punto di accesso con l'endpoint del punto di accesso

  Innanzitutto, per creare un endpoint S3 con il bucket incluso come parte del nome host, imposta lo stile di indirizzamento su `virtual` per `aws s3api`. Per ulteriori informazioni su `AWS configure`, consulta File di configurazione e delle credenziali nella Guida per l'utente di AWS Command Line Interface .

  ```
  aws configure set default.s3.addressing_style virtual
  ```

  Quindi, nell'esempio seguente, sostituisci l'alias del punto di accesso *accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias*, la Regione *us-east-1* e l'ID endpoint VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con le informazioni appropriate. Per ulteriori informazioni sull'alias del punto di accesso, consultaUtilizzo di un alias in stile bucket per il punto di accesso del bucket 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
```

Esempio: utilizzo dell'URL dell'endpoint per elencare i processi con un'operazione API di controllo S3

Nell'esempio seguente, sostituisci la Regione *us-east-1*, l'ID endpoint VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* e l'ID account *12345678* con le informazioni appropriate.

```
aws s3control --region us-east-1 --endpoint-url
 https://control.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com list-jobs --
account-id 12345678
```

AWS Esempi SDK

Per accedere ai bucket S3, agli access point S3 o alle operazioni dell'API Amazon S3 Control tramite gli endpoint dell'interfaccia S3 quando usi gli SDK, aggiorna i tuoi AWS SDK alla versione più recente. Quindi configura i client per utilizzare un URL endpoint per accedere a un bucket, un punto di accesso o un'operazione API di controllo Amazon S3 tramite gli endpoint di interfaccia S3.

SDK for Python (Boto3)

Esempio: utilizzo di un URL endpoint per accedere a un bucket S3

Nell'esempio seguente, sostituisci la Regione *us-east-1* e l'ID endpoint VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con le informazioni appropriate.

```
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'
)
```

Esempio: utilizzo di un URL endpoint per accedere a un access point S3

Nell'esempio seguente, sostituisci la Regione *us-east-1* e l'ID endpoint VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con le informazioni appropriate.

```
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'
)
```

Esempio: utilizzo di un URL endpoint per accedere all'API di controllo Amazon S3

Nell'esempio seguente, sostituisci la Regione *us-east-1* e l'ID endpoint VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con le informazioni appropriate.

```
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

Esempio: utilizzo di un URL endpoint per accedere a un bucket S3

Nell'esempio seguente, sostituisci l'ID endpoint VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con le informazioni appropriate.

```
// 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();
```

Esempio: utilizzo di un URL endpoint per accedere a un access point S3

Nell'esempio seguente, sostituisci l'ID endpoint VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* e l'ARN *us-east-1:123456789012:accesspoint/prod* con le informazioni appropriate.

```
// 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");
```

Esempio: utilizzo di un URL endpoint per accedere all'operazione API di controllo Amazon S3

Nell'esempio seguente, sostituisci l'ID endpoint VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con le informazioni appropriate.

```
// 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

Esempio: utilizzo di un URL endpoint per accedere a un bucket S3

Nell'esempio seguente, sostituisci l'ID endpoint VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* e la Regione *Region.US_EAST_1* con le informazioni appropriate.

```
// 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()
```

Esempio: utilizzo di un URL endpoint per accedere a un access point S3

Nell'esempio seguente, sostituisci l'ID endpoint VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* e la Regione *Region.US_EAST_1* con le informazioni appropriate.

```
// 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()
```

Esempio: utilizzo di un URL endpoint per accedere all'API di controllo Amazon S3

Nell'esempio seguente, sostituisci l'ID endpoint VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* e la Regione *Region.US_EAST_1* con le informazioni appropriate.

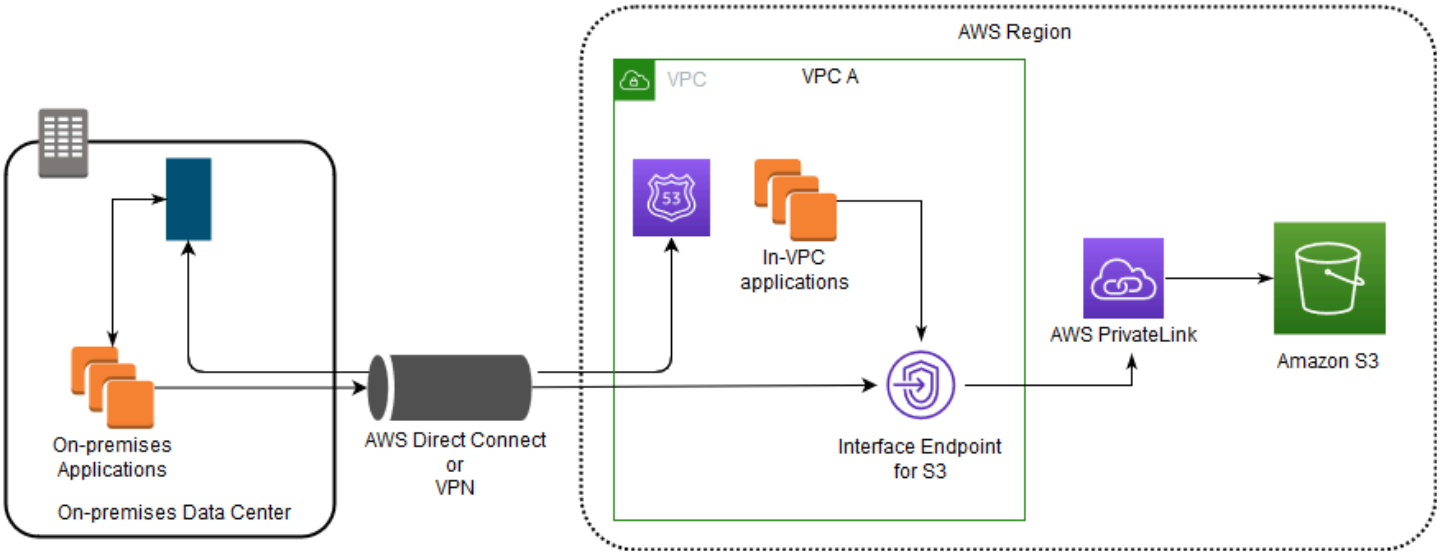
```
// control client
Region region = Region.US_EAST_1;
s3ControlClient = S3ControlClient.builder().region(region)

 .endpointOverride(URI.create("https://control.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com"))
                                  .build()
```

Aggiornamento di una configurazione DNS locale

Quando si utilizzano nomi DNS specifici degli endpoint per accedere agli endpoint di interfaccia per Amazon S3, non è necessario aggiornare il resolver DNS locale. Puoi risolvere il nome DNS specifico dell'endpoint con l'indirizzo IP privato dell'endpoint di interfaccia dal dominio DNS Amazon S3 pubblico.

Utilizzo degli endpoint di interfaccia per accedere ad Amazon S3 senza un endpoint gateway o un gateway Internet nel VPC

Gli endpoint di interfaccia nel VPC possono instradare sia le applicazioni nel VPC che le applicazioni locali ad Amazon S3 sulla rete Amazon, come illustrato nel diagramma seguente.

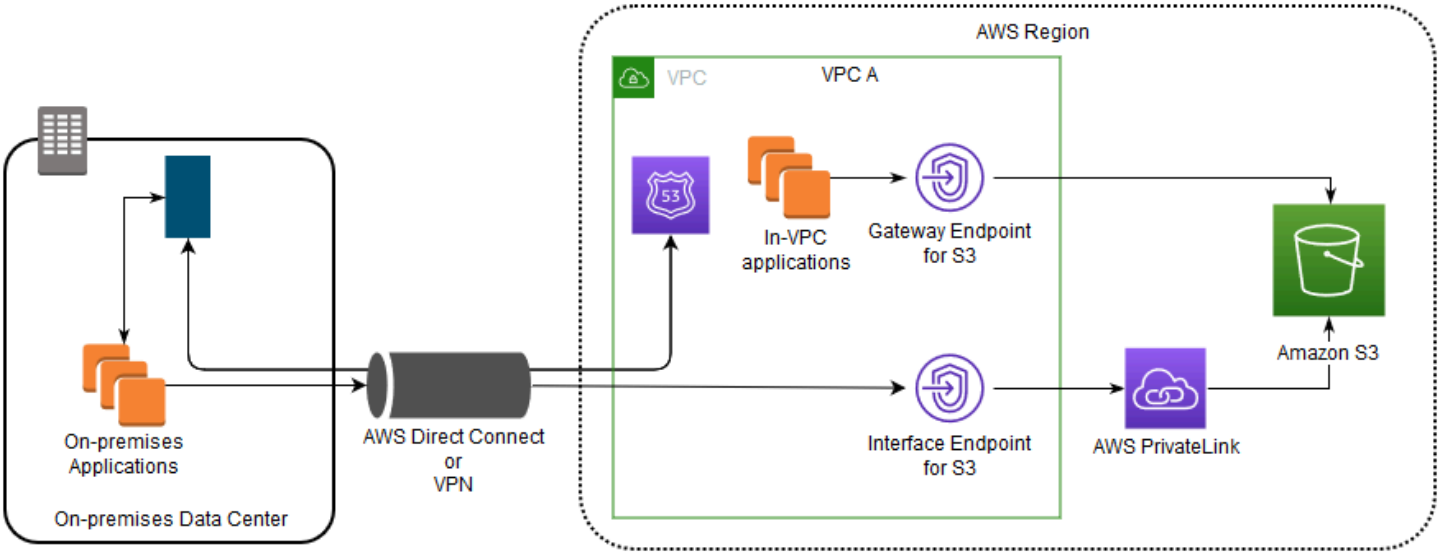


Il diagramma illustra quanto segue:

- La tua rete locale utilizza AWS Direct Connect o AWS VPN per connettersi a VPC A.
- Le applicazioni in locale e in VPC A utilizzano nomi DNS specifici degli endpoint per accedere ad Amazon S3 tramite l'endpoint di interfaccia S3.
- Le applicazioni locali inviano i dati all'endpoint di interfaccia nel VPC tramite AWS Direct Connect (o). AWS VPN AWS PrivateLink sposta i dati dall'endpoint dell'interfaccia ad Amazon S3 tramite AWS la rete.
- Le applicazioni in-VPC inviano inoltre traffico all'endpoint dell'interfaccia. AWS PrivateLink sposta i dati dall'endpoint dell'interfaccia ad Amazon S3 tramite AWS la rete.

Utilizzo di endpoint gateway e endpoint di interfaccia insieme nello stesso VPC per accedere ad Amazon S3

Puoi creare endpoint di interfaccia e mantenere l'endpoint gateway esistente nello stesso VPC, come illustrato nel diagramma seguente. Con questo approccio consenti alle applicazioni nel VPC di continuare ad accedere ad Amazon S3 tramite l'endpoint gateway senza essere fatturate. Quindi, solo le applicazioni on-premise utilizzerebbero gli endpoint di interfaccia per accedere ad Amazon S3. Per accedere ad Amazon S3 in questo modo, è necessario aggiornare le applicazioni On-Premise per utilizzare nomi DNS specifici degli endpoint per Amazon S3.



Il diagramma illustra quanto segue:

- Le applicazioni locali utilizzano nomi DNS specifici dell'endpoint per inviare dati all'endpoint di interfaccia all'interno del VPC tramite (o). AWS Direct Connect AWS VPN AWS PrivateLink sposta i dati dall'endpoint dell'interfaccia ad Amazon S3 tramite AWS la rete.
- Utilizzando i nomi regionali Amazon S3 predefiniti, le applicazioni in-VPC inviano dati all'endpoint gateway che si connette ad Amazon S3 tramite la rete. AWS

Per ulteriori informazioni sugli endpoint gateway, consulta Endpoint VPC gateway nella Guida per l'utente di VPC.

Creazione di una policy per l'endpoint VPC per Amazon S3

Puoi allegare una policy di endpoint all'endpoint VPC che controlla l'accesso ad Amazon S3. Questa policy specifica le informazioni riportate di seguito:

- Il principale AWS Identity and Access Management (IAM) che può eseguire azioni
- Le azioni che possono essere eseguite
- Le risorse sui cui si possono eseguire le azioni

Puoi utilizzare le policy del bucket di Amazon S3 anche per limitare l'accesso a bucket specifici da un endpoint VPC specifico utilizzando la condizione `aws:sourceVpce` nella policy del bucket. Negli esempi seguenti vengono illustrate le policy che limitano l'accesso a un bucket o a un endpoint.

Argomenti

Esempio: limitazione dell'accesso a un bucket specifico da un endpoint VPC

Puoi creare una policy di endpoint che limita l'accesso solo a bucket Amazon S3 specifici. Questo tipo di policy è utile se Servizi AWS nel tuo VPC sono presenti altre policy che utilizzano bucket. La seguente policy del bucket limita l'accesso solo a *example-s3-bucket1*. Per utilizzare questa policy di endpoint, sostituisci *example-s3-bucket1* con il nome del 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:::example-s3-bucket1",
                   "arn:aws:s3:::example-s3-bucket1/*"]
    }
  ]
}
```

Esempio: limitazione dell'accesso ai bucket in un account specifico da un endpoint VPC

Puoi creare una policy per gli endpoint che limiti l'accesso solo ai bucket S3 in uno specifico caso. Account AWS Per impedire ai client nel VPC di accedere ai bucket di cui non sei proprietario, utilizza la seguente istruzione nella policy di endpoint. Nell'esempio seguente viene creata una policy che limita l'accesso alle risorse di proprietà di un singolo ID Account 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**
>
> Per specificare l' Account AWS ID della risorsa a cui si accede, puoi utilizzare la `aws:ResourceAccount` o la `s3:ResourceAccount` chiave nella tua policy IAM. Tuttavia, tieni presente che alcuni Servizi AWS si basano sull'accesso ai bucket AWS gestiti. Pertanto, l'utilizzo della chiave di condizione `aws:ResourceAccount` o `s3:ResourceAccount` nella policy IAM potrebbe anche influire sull'accesso a tali risorse.

Esempio: limitazione dell'accesso a un endpoint VPC specifico nella policy del bucket S3

Esempio: limitazione dell'accesso a un endpoint VPC specifico nella policy del bucket S3

La seguente policy del bucket Amazon S3 consente l'accesso a un bucket specifico, *example-s3-bucket2*, solo dall'endpoint VPC *vpce-1a2b3c4d*. La policy nega l'accesso al bucket se l'endpoint specificato non è in uso. La condizione `aws:sourceVpce` viene utilizzata per specificare l'endpoint e non richiede un nome della risorsa Amazon (ARN) per la risorsa dell'endpoint VPC, ma solo l'ID dell'endpoint. Per utilizzare questa politica del bucket, sostituisci *example-s3-bucket2* e *vpce-1a2b3c4d* con il nome e l'endpoint del bucket.

Important

- Quando applichi la seguente policy del bucket Amazon S3 per limitare l'accesso solo a determinati endpoint VPC, potresti senza volerlo bloccare l'accesso al bucket. Le policy del bucket che hanno lo scopo di limitare l'accesso del bucket a connessioni originate dall'endpoint VPC possono bloccare tutte le connessioni al bucket. Per informazioni su come risolvere questo problema, consulta La policy del bucket ha l'ID del VPC o dell'endpoint VPC sbagliato. Come posso correggere la policy in modo da poter accedere al bucket? nel Knowledge Center di AWS Support .
- Prima di utilizzare la policy di esempio seguente, sostituire l'ID endpoint VPC con un valore appropriato per il caso d'uso. In caso contrario, non sarà possibile accedere al bucket.
- Questa policy disabilita l'accesso alla console al bucket specificato in quanto le richieste della console non provengono dall'endpoint VPC specificato.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909152",
  "Statement": [
    { "Sid": "Access-to-specific-VPCE-only",
      "Principal": "*",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": ["arn:aws:s3:::example-s3-bucket2",
                   "arn:aws:s3:::example-s3-bucket2/*"],
      "Condition": {"StringNotEquals": {"aws:sourceVpce": "vpce-1a2b3c4d"}}
    }
  ]
}
```

Per altri esempi di policy, consulta Endpoint per Amazon S3 nella Guida per l'utente di VPC.

Per ulteriori informazioni sulla connettività VPC, consulta le opzioni di connettività da rete a VPC nel white paper Opzioni di connettività AWS Amazon Virtual Private Cloud.

Gestione degli accessi

In AWS, una risorsa è un'entità con cui puoi lavorare. In Amazon Simple Storage Service (S3), i bucket e gli oggetti sono le risorse Amazon S3 originali. È probabile che ogni cliente S3 abbia dei bucket contenenti oggetti. Con l'aggiunta di nuove funzionalità a S3, sono state aggiunte anche risorse aggiuntive, ma non tutti i clienti utilizzano queste risorse specifiche per le funzionalità. Per ulteriori informazioni sulle risorse di Amazon S3, consulta. Risorse S3

Per impostazione predefinita, tutte le risorse Amazon S3 sono private. Per impostazione predefinita, l'utente root di chi ha creato la Account AWS risorsa (proprietario della risorsa) e gli utenti IAM all'interno di quell'account con le autorizzazioni necessarie possono accedere a una risorsa da loro creata. Il proprietario della risorsa decide chi altro può accedere alla risorsa e le azioni che altri possono eseguire sulla risorsa. S3 dispone di vari strumenti di gestione degli accessi che puoi utilizzare per concedere ad altri l'accesso alle tue risorse S3.

Le sezioni seguenti forniscono una panoramica delle risorse S3, degli strumenti di gestione degli accessi S3 disponibili e dei migliori casi d'uso per ogni strumento di gestione degli accessi. Gli elenchi contenuti in queste sezioni vogliono essere completi e includono tutte le risorse S3, gli strumenti di gestione degli accessi e i casi d'uso comuni di gestione degli accessi. Allo stesso tempo, queste sezioni sono progettate per essere directory che consentono di accedere ai dettagli tecnici desiderati. Se hai una buona conoscenza di alcuni dei seguenti argomenti, puoi passare alla sezione che ti riguarda.

Argomenti

- Verifica della proprietà del bucket con condizione del proprietario del bucket
- Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket

Risorse S3

Le risorse originali di Amazon S3 sono i bucket e gli oggetti in essi contenuti. Man mano che vengono aggiunte nuove funzionalità a S3, vengono aggiunte anche nuove risorse. Di seguito è riportato un elenco completo delle risorse S3 e delle rispettive funzionalità.

Tipo di risorsa	Funzionalità Amazon S3	Descrizione
`bucket`	Caratteristiche principali	Un bucket è un container per oggetti o file. Per memorizzare un oggetto in S3, crea un bucket e poi carica uno o più oggetti nel bucket. Per ulteriori informazioni, consulta Creazione, configurazione e utilizzo di bucket Amazon S3.
`object`		Un oggetto può essere un file e qualsiasi metadato che descrive quel file. Quando un oggetto è nel bucket, puoi aprirlo, scaricarlo e spostarlo. Per ulteriori informazioni, consulta Caricamento, download e utilizzo di oggetti in Amazon S3.
`accesspoint`	Access point	Gli access point sono endpoint di rete denominati collegati a bucket che è possibile utilizzare per eseguire operazioni sugli oggetti Amazon S3, ad esempio `GetObject` e `PutObject`. Ogni punto di accesso dispone di autorizzazioni e controlli di rete distinti e di una politica personalizzata del punto di accesso che funziona in combinazione con la policy del bucket associata al bucket sottostante. Puoi configurare qualsiasi punto di accesso per accettare richieste solo da un cloud privato virtuale (VPC) o configurare impostazioni di accesso pubblico a blocchi personalizzate per ogni punto di accesso. Per ulteriori informazioni, consulta Gestione dell'accesso ai dati con Punti di accesso Amazon S3.

Tipo di risorsa	Funzionalità Amazon S3	Descrizione
`objectlambdaaccesspoint`		Un Object Lambda Access Point è un punto di accesso per un bucket associato anche a una funzione Lambda. Con Object Lambda Access Point, puoi aggiungere il tuo codice ad Amazon `GET` S3 `LIST` `HEAD` e richiedere di modificare ed elaborare i dati non appena vengono restituiti a un'applicazione. Per ulteriori informazioni, consulta Creazione di punti di accesso Object Lambda.
`multiregionaccesspoint`		I punti di accesso multiregione forniscono un endpoint globale che le applicazioni possono utilizzare per soddisfare le richieste dei bucket Amazon S3 che si trovano in più regioni. AWS Puoi utilizzare i punti di accesso multi-regione per creare applicazioni multi-regione con la stessa architettura utilizzata in una singola regione e quindi eseguire tali applicazioni in qualsiasi parte del mondo. Invece di inviare richieste sulla rete Internet pubblica congestionata, le richieste di applicazioni effettuate a un endpoint globale Multi-Region Access Point vengono instradate automaticamente attraverso la rete AWS globale fino al bucket Amazon S3 più vicino. Per ulteriori informazioni, consulta Punti di accesso multi-regione in Amazon S3.
`job`	Operazioni in batch S3	Un job è una risorsa della funzionalità Batch Operations di S3. Puoi utilizzare S3 Batch Operations per eseguire operazioni batch su larga scala su elenchi di oggetti Amazon S3 specificati. Amazon S3 monitora lo stato di avanzamento del processo operativo in batch, invia notifiche e archivia un rapporto dettagliato di completamento di tutte le azioni, offrendoti un'esperienza completamente gestita, verificabile e senza server. Per ulteriori informazioni, consulta Esecuzione di operazioni in batch su larga scala su oggetti Amazon S3.

Tipo di risorsa	Funzionalità Amazon S3	Descrizione
`storagelensconfiguration`	S3 Storage Lens	Una configurazione S3 Storage Lens raccoglie i parametri di storage e i dati utente a livello di organizzazione su tutti gli account. S3 Storage Lens offre agli amministratori una visione unica dell'utilizzo e dell'attività dello storage di oggetti su centinaia o addirittura migliaia di account di un'organizzazione, con dettagli per generare approfondimenti a più livelli di aggregazione. Per ulteriori informazioni, consulta Valutazione dell'attività e dell'utilizzo dello storage con Amazon S3 Storage Lens.
`storagelensgroup`		Un gruppo S3 Storage Lens aggrega le metriche utilizzando filtri personalizzati basati sui metadati degli oggetti. I gruppi S3 Storage Lens ti aiutano a studiare le caratteristiche dei tuoi dati, come la distribuzione degli oggetti per età, i tipi di file più comuni e altro ancora. Per ulteriori informazioni, consulta Utilizzo dei gruppi S3 Storage Lens.
`accessgrantsinstance`	S3 Access Grants	Un'istanza S3 Access Grants è un contenitore per le concessioni S3 che crei. Con S3 Access Grants, puoi creare sovvenzioni ai tuoi dati Amazon S3 per identità IAM all'interno del tuo account, identità IAM in altri account (più account) e identità di directory aggiunte dalla tua directory aziendale . AWS IAM Identity Center Per ulteriori informazioni su S3 Access Grants, consulta. Gestione dell'accesso con S3 Access Grants

Tipo di risorsa	Funzionalità Amazon S3	Descrizione
`accessgrantslocation`		Un Access Grants Location è un bucket, un prefisso all'interno di un bucket o un oggetto che registri nell'istanza di S3 Access Grants. È necessario registrare le sedi all'interno dell'istanza S3 Access Grants prima di poter creare una concessione per quella posizione. Quindi, con S3 Access Grants, puoi concedere l'accesso al bucket, al prefisso o all'oggetto per le identità IAM all'interno del tuo account, le identità IAM in altri account (più account) e le identità di directory aggiunte dalla tua directory aziendale. AWS IAM Identity Center Per ulteriori informazioni su S3 Access Grants, vedi Gestione dell'accesso con S3 Access Grants
`accessgrant`		Una concessione di accesso è una concessione individuale ai tuoi dati Amazon S3. Con S3 Access Grants, puoi creare sovvenzioni ai tuoi dati Amazon S3 per identità IAM all'interno del tuo account, identità IAM in altri account (più account) e identità di directory aggiunte dalla tua directory aziendale . AWS IAM Identity Center Per ulteriori informazioni su S3 Access Grants, consulta Gestione dell'accesso con S3 Access Grants

Bucket

Esistono due tipi di bucket Amazon S3: bucket generici e bucket di directory.

- I bucket per uso generico sono il tipo di bucket S3 originale e sono consigliati per la maggior parte dei casi d'uso e dei modelli di accesso. I bucket per uso generico consentono l'uso di oggetti archiviati in tutte le classi di archiviazione, fatta eccezione per S3 Express One Zone. Per ulteriori informazioni sulle classi di storage S3, consulta. Utilizzo delle classi di storage di Amazon S3
- I bucket di directory utilizzano la classe di storage S3 Express One Zone, consigliata se l'applicazione è sensibile alle prestazioni e sfrutta i millisecondi e le latenze a una cifra. `PUT` `GET` Per ulteriori informazioni, consulta Bucks di directory, Che cos'è S3 Express One Zone? e AWS Identity and Access Management (IAM) per S3 Express One Zone.

Classificazione delle risorse S3

Amazon S3 offre funzionalità per classificare e organizzare le risorse S3. La categorizzazione delle risorse non è utile solo per organizzarle, ma consente anche di impostare regole di gestione degli accessi basate sulle categorie di risorse. In particolare, i prefissi e i tag sono due funzionalità di organizzazione dell'archiviazione che è possibile utilizzare per impostare le autorizzazioni di gestione degli accessi.

> **Note**
>
> Le seguenti informazioni si applicano ai bucket per uso generico. I bucket di directory non supportano l'etichettatura e presentano limitazioni relative ai prefissi. Per ulteriori informazioni, consulta AWS Identity and Access Management (IAM) per S3 Express One Zone.

- Prefissi: un prefisso in Amazon S3 è una stringa di caratteri all'inizio del nome chiave di un oggetto che viene utilizzata per organizzare gli oggetti archiviati nei bucket S3. È possibile utilizzare un carattere delimitatore, ad esempio una barra (/), per indicare la fine del prefisso all'interno del nome della chiave dell'oggetto. Ad esempio, potreste avere nomi di chiavi di oggetto che iniziano con il `engineering/` prefisso o nomi di chiavi di oggetto che iniziano con il prefisso. `marketing/campaigns/` L'uso di un delimitatore alla fine del prefisso, ad esempio una barra, / emula le convenzioni di denominazione di cartelle e file. Tuttavia, in S3, il prefisso fa parte del nome della chiave dell'oggetto. Nei bucket S3 per uso generico, non esiste una vera gerarchia di cartelle.

  Amazon S3 supporta l'organizzazione e il raggruppamento di oggetti utilizzando i relativi prefissi. Puoi anche gestire l'accesso agli oggetti tramite i relativi prefissi. Ad esempio, è possibile limitare l'accesso solo agli oggetti con nomi che iniziano con un prefisso specifico.

  Per ulteriori informazioni, consulta Organizzazione degli oggetti utilizzando i prefissi. La console S3 utilizza il concetto di cartelle, che, nei bucket generici, sono essenzialmente prefissi aggiunti al nome della chiave dell'oggetto. Per ulteriori informazioni, consulta Organizzazione degli oggetti nella console di Amazon S3 utilizzando le cartelle.
- Tag: ogni tag è una coppia chiave-valore che assegni alle risorse. Ad esempio, puoi taggare alcune risorse con il tag. `topicCategory=engineering` È possibile utilizzare i tag per facilitare l'allocazione dei costi, la categorizzazione e l'organizzazione e il controllo degli accessi. Il bucket tagging viene utilizzato solo per l'allocazione dei costi. Puoi taggare oggetti, S3 Storage Lens, lavori e S3 Access Grants per scopi organizzativi o per il controllo degli accessi. In S3

Access Grants, puoi anche utilizzare i tag per l'allocazione dei costi. Come esempio di controllo dell'accesso alle risorse utilizzando i relativi tag, puoi condividere solo gli oggetti che hanno un tag specifico o una combinazione di tag.

Per ulteriori informazioni, consulta Controllare l'accesso alle AWS risorse utilizzando i tag delle risorse nella Guida per l'utente IAM.

Identità

In Amazon S3, il proprietario della risorsa è l'identità che ha creato la risorsa, ad esempio un bucket o un oggetto. Per impostazione predefinita, solo l'utente root dell'account che ha creato la risorsa e le identità IAM all'interno dell'account che dispongono dell'autorizzazione richiesta possono accedere alla risorsa S3. I proprietari delle risorse possono consentire ad altre identità di accedere alle proprie risorse S3.

Le identità che non possiedono una risorsa possono richiedere l'accesso a tale risorsa. Le richieste a una risorsa sono autenticate o non autenticate. Le richieste autenticate devono includere un valore di firma che autentichi il mittente della richiesta, ma le richieste non autenticate non richiedono una firma. Si consiglia di concedere l'accesso solo agli utenti autenticati. Per ulteriori informazioni sull'autenticazione delle richieste, consulta Esecuzione di richieste.

> **Important**
>
> Ti consigliamo di non utilizzare le credenziali dell'utente Account AWS root per effettuare richieste autenticate. Crea invece un ruolo IAM, concedendo a esso l'accesso completo. Gli utenti con questo ruolo vengono definiti utenti amministratori. È possibile utilizzare le credenziali assegnate al ruolo di amministratore, anziché le credenziali dell'utente Account AWS root, per interagire AWS ed eseguire attività, come creare un bucket, creare utenti e concedere autorizzazioni. Per ulteriori informazioni, consulta le credenziali dell'utente Account AWS root e le credenziali utente IAM in Riferimenti generali di AWS, e consulta le migliori pratiche di sicurezza in IAM nella IAM User Guide.

Le identità che accedono ai tuoi dati in Amazon S3 possono essere una delle seguenti:

Account AWS owner

Account AWS Quello che ha creato la risorsa. Ad esempio, l'account che ha creato il bucket. Questo account possiede la risorsa. Per ulteriori informazioni, vedere AWS account root user.

Identità IAM nello stesso account del proprietario Account AWS

Quando configura gli account per i nuovi membri del team che richiedono l'accesso a S3, il Account AWS proprietario può utilizzare AWS Identity and Access Management (IAM) per creare utenti, gruppi e ruoli. Il Account AWS proprietario può quindi condividere le risorse con queste identità IAM. Il proprietario dell'account può anche specificare le autorizzazioni da concedere alle identità IAM, che consentono o negano le azioni che possono essere eseguite sulle risorse condivise.

Le identità IAM offrono funzionalità avanzate, inclusa la possibilità di richiedere agli utenti di inserire le credenziali di accesso prima di accedere alle risorse condivise. Utilizzando le identità IAM, puoi implementare una forma di autenticazione a più fattori (MFA) IAM per supportare una solida base di identità. Una best practice di IAM consiste nel creare ruoli per la gestione degli accessi anziché concedere le autorizzazioni a ogni singolo utente. Assegni ai singoli utenti il ruolo appropriato. Per ulteriori informazioni, consulta Best practice per la sicurezza in IAM.

Altri proprietari di AWS account e relative identità IAM (accesso tra account diversi)

Il Account AWS proprietario può inoltre consentire ad altri proprietari di AWS account, o identità IAM che appartengono a un altro AWS account, l'accesso alle risorse.

> **Note**
>
> Delega delle autorizzazioni: se un utente Account AWS possiede una risorsa, può concedere tali autorizzazioni a un'altra. Account AWS Tale account può quindi delegare tali autorizzazioni, o un sottoinsieme di esse, agli utenti dello stesso account. Questa operazione si definisce delega delle autorizzazioni. Tuttavia, un account che riceve le autorizzazioni da un altro account non può delegare tali autorizzazioni «su più account» a un altro. Account AWS

Utenti anonimi (accesso pubblico)

Il Account AWS proprietario può rendere pubbliche le risorse. Rendendo pubblica una risorsa tecnicamente la risorsa viene condivisa con l'utente anonimo. I bucket creati a partire da aprile 2023 bloccano tutti gli accessi pubblici per impostazione predefinita, a meno che non si modifichi questa impostazione. Ti consigliamo di impostare i bucket per bloccare l'accesso pubblico e di concedere l'accesso solo agli utenti autenticati. Per ulteriori informazioni sul blocco dell'accesso pubblico, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

Servizi AWS

Il proprietario della risorsa può concedere a un altro AWS servizio l'accesso a una risorsa Amazon S3. Ad esempio, puoi concedere al AWS CloudTrail servizio l'`s3:PutObject`autorizzazione a scrivere file di registro nel tuo bucket. Per ulteriori informazioni, vedere Fornire l'accesso a un AWS servizio.

Identità degli elenchi aziendali

Il proprietario della risorsa può concedere agli utenti o ai ruoli della directory aziendale l'accesso a una risorsa S3 utilizzando S3 Access Grants. Per ulteriori informazioni sull'aggiunta della directory aziendale a AWS IAM Identity Center, consulta Cos'è IAM Identity Center? .

Proprietari di bucket o risorse

I bucket Account AWS che usi per creare bucket e caricare oggetti sono i proprietari di tali risorse. Il proprietario di un bucket può concedere autorizzazioni su più account a un altro Account AWS (o agli utenti di un altro account) per caricare oggetti.

Quando il proprietario del bucket consente a un altro account di caricare oggetti in un bucket, il proprietario del bucket, per impostazione predefinita, possiede tutti gli oggetti caricati nel proprio bucket. Tuttavia, se le impostazioni preferite del bucket proprietario del bucket e del proprietario del bucket sono disattivate, chi carica gli oggetti possiede tali oggetti e il Account AWS proprietario del bucket non dispone delle autorizzazioni sugli oggetti di proprietà di un altro account, con le seguenti eccezioni:

- È il proprietario del bucket a pagare la fattura. Il proprietario del bucket può rifiutare l'accesso agli oggetti nel bucket o eliminarli, indipendentemente dall'utente a cui appartengono.
- Il proprietario del bucket può archiviare qualsiasi oggetto o ripristinare gli oggetti archiviati, indipendentemente dal proprietario. L'archiviazione fa riferimento alla classe di storage utilizzata per archiviare gli oggetti. Per ulteriori informazioni, consulta Gestione del ciclo di vita dello storage.

Strumenti di gestione degli accessi

Amazon S3 offre una varietà di funzionalità e strumenti di sicurezza. Di seguito è riportato un elenco completo di queste funzionalità e strumenti. Non sono necessari tutti questi strumenti di gestione degli accessi, ma è necessario utilizzarne uno o più per concedere l'accesso alle risorse Amazon S3. La corretta applicazione di questi strumenti può contribuire a garantire che le risorse siano accessibili solo agli utenti previsti.

Lo strumento di gestione degli accessi più utilizzato è una politica di accesso. Una politica di accesso può essere una politica basata sulle risorse collegata a una AWS risorsa, ad esempio una policy bucket per un bucket. Una policy di accesso può anche essere una policy basata sull'identità collegata a un'identità AWS Identity and Access Management (IAM), ad esempio un utente, un gruppo o un ruolo IAM. Scrivi una policy di accesso per concedere a utenti, gruppi Account AWS e ruoli IAM l'autorizzazione a eseguire operazioni su una risorsa. Ad esempio, puoi concedere `PUT Object` l'autorizzazione a un altro account Account AWS in modo che l'altro account possa caricare oggetti nel tuo bucket.

Una politica di accesso descrive chi ha accesso a quali cose. Quando Amazon S3 riceve una richiesta, deve valutare tutte le politiche di accesso per determinare se autorizzare o rifiutare la richiesta. Per ulteriori informazioni su come Amazon S3 valuta le policy, consulta In che modo Amazon S3 autorizza una richiesta.

Di seguito sono riportati gli strumenti di gestione degli accessi disponibili in Amazon S3.

Policy del bucket

Una policy sui bucket di Amazon S3 è una policy basata su risorse in formato JSON AWS Identity and Access Management (IAM) collegata a un particolare bucket. Utilizza le policy dei bucket per concedere autorizzazioni ad altre identità Account AWS o IAM per il bucket e gli oggetti in esso contenuti. Molti casi d'uso della gestione degli accessi di S3 possono essere soddisfatti utilizzando una bucket policy. Con le bucket policy, puoi personalizzare l'accesso ai bucket per assicurarti che solo le identità che hai approvato possano accedere alle risorse ed eseguire azioni al loro interno. Per ulteriori informazioni, consulta Politiche Bucket per Amazon S3.

Di seguito è riportato un esempio di policy di bucket. La policy del bucket viene espressa utilizzando un file JSON. Questa policy di esempio concede l'autorizzazione di lettura del ruolo IAM a tutti gli oggetti nel bucket. Contiene un'istruzione denominata`BucketLevelReadPermissions`, che consente l'`s3:GetObject`azione (autorizzazione di lettura) sugli oggetti in un bucket denominato. `DOC-EXAMPLE-BUCKET1` Specificando un ruolo IAM come`Principal`, questa policy concede l'accesso a qualsiasi utente IAM con questo ruolo. Per utilizzare questa policy di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"BucketLevelReadPermissions",
```

```
      "Effect":"Allow",
      "Principal": {
     "AWS": "arn:aws:iam::123456789101:role/s3-role"
      },
      "Action":["s3:GetObject"],
      "Resource":["arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*"]
    }]
  }
```

> **Note**
>
> Durante la creazione delle policy, è opportuno evitare l'uso di caratteri jolly (*) nell'elemento `Principal` perché questo consente a chiunque di accedere alle risorse Amazon S3. Invece, elenca in modo esplicito gli utenti o i gruppi a cui è consentito accedere al bucket o elenca le condizioni che devono essere soddisfatte utilizzando una clausola condizionale nella policy. Inoltre, anziché includere un carattere jolly per le azioni degli utenti o dei gruppi, concedi loro autorizzazioni specifiche quando applicabile.

Policy basata su identità

Una policy utente basata sull'identità o IAM è un tipo di AWS Identity and Access Management policy (IAM). Una policy basata sull'identità è una policy in formato JSON associata agli utenti, ai gruppi o ai ruoli IAM del tuo account. AWS Puoi utilizzare policy basate sull'identità per concedere a un'identità IAM l'accesso ai tuoi bucket o oggetti. Puoi creare utenti, gruppi e ruoli IAM nel tuo account e allegare loro policy di accesso. Puoi quindi concedere l'accesso alle AWS risorse, incluse le risorse Amazon S3. Per ulteriori informazioni, consulta Policy basate sull'identità per Amazon S3.

Di seguito è riportato un esempio di policy basata sull'identità. La policy di esempio consente al ruolo IAM associato di eseguire sei diverse azioni Amazon S3 (autorizzazioni) su un bucket e sugli oggetti in esso contenuti. Se colleghi questa policy a un ruolo IAM nel tuo account e assegni il ruolo ad alcuni dei tuoi utenti IAM, gli utenti con questo ruolo saranno in grado di eseguire queste azioni sulle risorse (bucket) specificate nella tua policy. Per utilizzare questa policy di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Sid": "AssignARoleActions",
```

```
    "Effect": "Allow",
    "Action": [
   "s3:PutObject",
   "s3:GetObject",
   "s3:ListBucket",
   "s3:DeleteObject",
   "s3:GetBucketLocation"
    ],
    "Resource": [
   "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*",
   "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
    ]
    },
  {
    "Sid": "AssignARoleActions2",
    "Effect": "Allow",
    "Action": "s3:ListAllMyBuckets",
    "Resource": "*"
  }
  ]
}
```

S3 Access Grants

Usa S3 Access Grants per creare concessioni di accesso ai tuoi dati Amazon S3 per entrambe le identità nelle directory di identità aziendali, ad esempio, e alle identità (IAM). Active Directory AWS Identity and Access Management S3 Access Grants ti aiuta a gestire le autorizzazioni relative ai dati su larga scala. Inoltre, S3 Access Grants registra l'identità dell'utente finale e l'applicazione utilizzata per accedere ai dati S3. AWS CloudTrail Ciò fornisce una cronologia di controllo dettagliata fino all'identità dell'utente finale per tutti gli accessi ai dati nei bucket S3. Per ulteriori informazioni, consulta Gestione dell'accesso con S3 Access Grants.

Access point

Amazon S3 Access Points semplifica la gestione dell'accesso ai dati su larga scala per le applicazioni che utilizzano set di dati condivisi su S3. Gli access point sono endpoint di rete denominati collegati a un bucket. È possibile utilizzare i punti di accesso per eseguire operazioni sugli oggetti S3 su larga scala, come il caricamento e il recupero di oggetti. Un bucket può avere fino a 10.000 punti di accesso collegati e, per ogni punto di accesso, puoi applicare autorizzazioni e controlli di rete distinti per avere un controllo dettagliato sull'accesso agli oggetti S3. Gli access point S3 possono essere associati a bucket nello stesso account o in un altro account affidabile. Le policy degli Access Points

sono politiche basate sulle risorse che vengono valutate insieme alla policy bucket sottostante. Per ulteriori informazioni, consulta Gestione dell'accesso ai dati con Punti di accesso Amazon S3.

Lista di controllo degli accessi (ACL)

Un ACL è un elenco di sovvenzioni che identificano il beneficiario e l'autorizzazione concessa. Gli ACL concedono autorizzazioni di base di lettura o scrittura ad altri. Account AWS Le liste ACL utilizzano uno schema XML specifico di Amazon S3. Un ACL è un tipo di policy AWS Identity and Access Management (IAM). Un ACL di oggetto viene utilizzato per gestire l'accesso a un oggetto e un ACL di un bucket viene utilizzato per gestire l'accesso a un bucket. Con le policy dei bucket, esiste un'unica politica per l'intero bucket, ma gli ACL degli oggetti sono specificati per ogni oggetto. Si consiglia di mantenere gli ACL disattivati, tranne in circostanze insolite in cui è necessario controllare singolarmente l'accesso per ciascun oggetto. Per ulteriori informazioni sulle ACL, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

> **Warning**
>
> La maggior parte dei casi d'uso moderni in Amazon S3 non richiede l'uso di ACL.

Di seguito è riportato un esempio di ACL del bucket. La concessione nell'ACL mostra un proprietario del bucket che dispone dell'autorizzazione al pieno controllo.

```
<?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>
```

Proprietà dell'oggetto

Per gestire l'accesso ai tuoi oggetti, devi essere il proprietario dell'oggetto. Puoi utilizzare l'impostazione a livello di bucket Object Ownership per controllare la proprietà degli oggetti caricati nel tuo bucket. Inoltre, usa Object Ownership per attivare gli ACL. Per impostazione predefinita, Object Ownership è impostata sull'impostazione applicata dal proprietario del Bucket e tutti gli ACL sono disattivati. Quando gli ACL sono disattivati, il proprietario del bucket possiede tutti gli oggetti nel bucket e gestisce esclusivamente l'accesso ai dati. Per gestire l'accesso, il proprietario del bucket utilizza le policy o un altro strumento di gestione degli accessi, esclusi gli ACL. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

Object Ownership dispone di tre impostazioni che puoi utilizzare sia per controllare la proprietà degli oggetti caricati nel tuo bucket sia per attivare gli ACL:

ACL disattivati

- Proprietario del bucket applicato (impostazione predefinita): gli ACL sono disattivati e il proprietario del bucket possiede automaticamente e ha il pieno controllo su ogni oggetto nel bucket. Gli ACL non influiscono sulle autorizzazioni per i dati nel bucket S3. Il bucket utilizza le policy esclusivamente per definire il controllo degli accessi.

Gli ACL sono attivati

- Proprietario del bucket scelto – Il proprietario del bucket possiede e ha il pieno controllo sui nuovi oggetti che altri account scrivono nel bucket con l'ACL predefinita `bucket-owner-full-control`.
- Scrittore di oggetti: chi carica un oggetto possiede l'oggetto, ne ha il pieno controllo e può concedere ad altri utenti l'accesso ad esso tramite ACL. Account AWS

Best practice aggiuntive

Prendi in considerazione l'utilizzo delle seguenti impostazioni e strumenti del bucket per proteggere i dati in transito e a riposo, entrambi fondamentali per mantenere l'integrità e l'accessibilità dei tuoi dati:

- Blocca accesso pubblico: non disattivare l'impostazione predefinita a livello di bucket Blocca accesso pubblico. Per impostazione predefinita, questa impostazione blocca l'accesso pubblico ai dati. Per ulteriori informazioni sul blocco dell'accesso pubblico, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

- S3 Versioning: per l'integrità dei dati, puoi implementare l'impostazione del bucket S3 Versioning, che crea una versione degli oggetti man mano che apporti aggiornamenti, anziché sovrascriverli. Puoi usare S3 Versioning per conservare, recuperare e ripristinare una versione precedente, se necessario. Per informazioni sulla funzione Controllo delle versioni S3, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.
- S3 Object Lock — S3 Object Lock è un'altra impostazione che puoi implementare per raggiungere l'integrità dei dati. Questa funzionalità può implementare un modello write-once-read-many (WORM) per archiviare oggetti in modo immutabile. Per ulteriori informazioni sul blocco oggetti, consulta Utilizzo del blocco oggetti S3.
- Crittografia degli oggetti: Amazon S3 offre diverse opzioni di crittografia degli oggetti che proteggono i dati in transito e a riposo. La crittografia lato server crittografa l'oggetto prima di salvarlo sui dischi dei relativi data center e quindi lo decrittografa quando gli oggetti vengono scaricati. Se autentichi la richiesta e disponi delle autorizzazioni di accesso, non c'è differenza nel modo in cui accedi agli oggetti crittografati o non crittografati. Per ulteriori informazioni, consulta Protezione dei dati con la crittografia lato server. Per impostazione predefinita, S3 crittografa gli oggetti appena caricati. Per ulteriori informazioni, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3. La crittografia lato client consiste nel crittografare i dati prima di inviarli ad Amazon S3. Per ulteriori informazioni, consulta Protezione dei dati con la crittografia lato client.
- Metodi di firma: la versione 4 di Signature è il processo di aggiunta di informazioni di autenticazione alle AWS richieste inviate tramite HTTP. Per motivi di sicurezza, la maggior parte delle richieste AWS deve essere firmata con una chiave di accesso, che consiste in un ID della chiave di accesso e una chiave di accesso segreta. Queste due chiavi in genere vengono definite come le tue credenziali di sicurezza. Per ulteriori informazioni, consulta le sezioni Autenticazione delle richieste (AWS Signature Version 4) e Processo di firma Signature Version 4.

Azioni

Per un elenco completo delle autorizzazioni e delle chiavi di condizione S3, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.

Azioni

Le azioni AWS Identity and Access Management (IAM) per Amazon S3 sono le possibili azioni che possono essere eseguite su un bucket o un oggetto S3. Concedi queste azioni alle identità in modo che possano agire sulle tue risorse S3. Esempi di azioni S3 sono la lettura `s3:GetObject` di oggetti in un bucket e `s3:PutObject` la scrittura di oggetti in un bucket.

Chiavi di condizione

Oltre alle azioni, le chiavi delle condizioni IAM si limitano a concedere l'accesso solo quando viene soddisfatta una condizione. I tasti di condizione sono opzionali.

> **Note**
>
> In una politica di accesso basata sulle risorse, ad esempio una policy sui bucket, o in una politica basata sull'identità, puoi specificare quanto segue:
>
> - Un'azione o una serie di azioni nell'elemento della dichiarazione politica. `Action`
> - Nell'`Effect`elemento della dichiarazione politica, è possibile specificare di `Allow` concedere le azioni elencate oppure è possibile specificare di `Deny` bloccare le azioni elencate. Per mantenere ulteriormente la pratica dei privilegi minimi, `Deny` le dichiarazioni nell'`Effect`elemento della politica di accesso devono essere le più ampie possibile e `Allow` le dichiarazioni devono essere il più ristrette possibile. `Deny`gli effetti associati all'`s3:*`azione sono un altro buon modo per implementare le migliori pratiche di opt-in per le identità incluse nelle dichiarazioni sulle condizioni politiche.
> - Una condizione chiave nell'`Condition`elemento di una dichiarazione politica.

Casi d'uso della gestione degli accessi

Amazon S3 offre ai proprietari delle risorse una varietà di strumenti per concedere l'accesso. Lo strumento di gestione degli accessi S3 che utilizzi dipende dalle risorse S3 che desideri condividere, dalle identità a cui concedi l'accesso e dalle azioni che desideri consentire o negare. Potresti voler utilizzare uno o una combinazione di strumenti di gestione degli accessi S3 per gestire l'accesso alle tue risorse S3.

Nella maggior parte dei casi, puoi utilizzare una politica di accesso per gestire le autorizzazioni. Una policy di accesso può essere una policy basata sulle risorse, collegata a una risorsa, ad esempio un bucket, o un'altra risorsa Amazon S3 (). Risorse S3 Una policy di accesso può anche essere una policy basata sull'identità, collegata a un utente, gruppo o ruolo AWS Identity and Access Management (IAM) nel tuo account. Potresti scoprire che una policy bucket funziona meglio per il tuo caso d'uso. Per ulteriori informazioni, consulta Politiche Bucket per Amazon S3. In alternativa, con AWS Identity and Access Management (IAM), puoi creare utenti, gruppi e ruoli IAM al tuo interno

Account AWS e gestire il loro accesso a bucket e oggetti tramite policy basate sull'identità. Per ulteriori informazioni, consulta Policy basate sull'identità per Amazon S3.

Per aiutarti a navigare tra queste opzioni di gestione degli accessi, di seguito sono riportati i casi d'uso e i consigli comuni dei clienti di Amazon S3 per ciascuno degli strumenti di gestione degli accessi S3.

Il Account AWS proprietario desidera condividere i bucket solo con utenti all'interno dello stesso account

Tutti gli strumenti di gestione degli accessi possono soddisfare questo caso d'uso di base. Consigliamo i seguenti strumenti di gestione degli accessi per questo caso d'uso:

- Policy sui bucket: se desideri concedere l'accesso a un bucket o a un numero limitato di bucket o se le tue autorizzazioni di accesso ai bucket sono simili da un bucket all'altro, utilizza una policy sui bucket. Con le policy dei bucket, gestisci una policy per ogni bucket. Per ulteriori informazioni, consulta Politiche Bucket per Amazon S3.
- Policy basata sull'identità: se disponi di un numero molto elevato di bucket con autorizzazioni di accesso diverse per ogni bucket e solo pochi ruoli utente da gestire, puoi utilizzare una policy IAM per utenti, gruppi o ruoli. Le policy IAM sono anche una buona opzione se gestisci l'accesso degli utenti ad altre AWS risorse, oltre alle risorse Amazon S3. Per ulteriori informazioni, consulta Esempio 1: il proprietario del bucket concede agli utenti le autorizzazioni per il bucket.
- S3 Access Grants: puoi utilizzare S3 Access Grants per concedere l'accesso ai tuoi bucket, prefissi o oggetti S3. S3 Access Grants consente di specificare diverse autorizzazioni a livello di oggetto su larga scala, mentre le policy dei bucket sono limitate a 20 KB di dimensione. Per ulteriori informazioni, consulta Nozioni di base su S3 Access Grants.
- Punti di accesso: è possibile utilizzare gli access point, denominati endpoint di rete collegati a un bucket. Un bucket può avere fino a 10.000 punti di accesso collegati e per ogni punto di accesso puoi applicare autorizzazioni e controlli di rete distinti per avere un controllo dettagliato sull'accesso ai tuoi oggetti S3. Per ulteriori informazioni, consulta Gestione dell'accesso ai dati con Punti di accesso Amazon S3.

Il Account AWS proprietario desidera condividere bucket o oggetti con utenti di un altro account (cross-account) AWS

Per concedere l'autorizzazione a un altro utente Account AWS, è necessario utilizzare una policy sui bucket o uno dei seguenti strumenti di gestione degli accessi consigliati. Non è possibile utilizzare

una politica di accesso basata sull'identità per questo caso d'uso. Per ulteriori informazioni sulla concessione dell'accesso a più account, consulta Come posso fornire l'accesso su più account agli oggetti che si trovano nei bucket Amazon S3?

Consigliamo i seguenti strumenti di gestione degli accessi per questo caso d'uso:

- Bucket policy: con le bucket policy, gestisci una policy per ogni bucket. Per ulteriori informazioni, consulta Politiche Bucket per Amazon S3.
- S3 Access Grants: puoi utilizzare S3 Access Grants per concedere autorizzazioni su più account ai tuoi bucket, prefissi o oggetti S3. Puoi utilizzare S3 Access Grants per specificare diverse autorizzazioni a livello di oggetto su larga scala, mentre le policy dei bucket hanno una dimensione limitata a 20 KB. Per ulteriori informazioni, consulta Nozioni di base su S3 Access Grants.
- Punti di accesso: puoi utilizzare i punti di accesso, denominati endpoint di rete collegati a un bucket. Un bucket può avere fino a 10.000 punti di accesso collegati e per ogni punto di accesso puoi applicare autorizzazioni e controlli di rete distinti per avere un controllo dettagliato sull'accesso ai tuoi oggetti S3. Per ulteriori informazioni, consulta Gestione dell'accesso ai dati con Punti di accesso Amazon S3.

Il Account AWS proprietario o il proprietario del bucket deve concedere le autorizzazioni a livello di oggetto o di prefisso e queste autorizzazioni variano da oggetto a oggetto o da prefisso a prefisso

In una policy bucket, ad esempio, puoi concedere l'accesso agli oggetti all'interno di un bucket che condividono un nome chiave specifico, un prefisso o un tag specifico. È possibile concedere l'autorizzazione di lettura agli oggetti che iniziano con il prefisso del nome chiave. `logs/` Tuttavia, se le autorizzazioni di accesso variano in base all'oggetto, la concessione delle autorizzazioni a singoli oggetti utilizzando una policy bucket potrebbe non essere pratica, soprattutto perché le policy dei bucket hanno una dimensione limitata a 20 KB.

Consigliamo i seguenti strumenti di gestione degli accessi per questo caso d'uso:

- S3 Access Grants: puoi utilizzare S3 Access Grants per gestire le autorizzazioni a livello di oggetto o prefisso. A differenza delle bucket policy, puoi usare S3 Access Grants per specificare autorizzazioni diverse a livello di oggetto su larga scala. Le policy di bucket sono limitate a una dimensione di 20 KB. Per ulteriori informazioni, consulta Nozioni di base su S3 Access Grants.
- Punti di accesso: puoi utilizzare i punti di accesso per gestire le autorizzazioni a livello di oggetto o prefisso. Gli access point sono denominati endpoint di rete collegati a un bucket. Un bucket può avere fino a 10.000 punti di accesso collegati e per ogni punto di accesso puoi applicare

autorizzazioni e controlli di rete distinti per avere un controllo dettagliato sull'accesso agli oggetti S3. Per ulteriori informazioni, consulta Gestione dell'accesso ai dati con Punti di accesso Amazon S3.

- ACL: non è consigliabile utilizzare gli Access Control List (ACL), soprattutto perché gli ACL sono limitati a 100 concessioni per oggetto. Tuttavia, se scegli di attivare gli ACL, nelle impostazioni del bucket imposta Object Ownership su Bucket owner favorite e ACL abilitati. Con questa impostazione, nuovi oggetti scritti con l'ACL predefinita `bucket-owner-full-control` saranno automaticamente di proprietà del proprietario del bucket anziché dell'object writer. Puoi quindi utilizzare gli ACL degli oggetti, che sono una politica di accesso in formato XML, per concedere ad altri utenti l'accesso all'oggetto. Per ulteriori informazioni, consulta Panoramica delle liste di controllo accessi (ACL).

Il Account AWS proprietario o il proprietario del bucket desidera limitare l'accesso al bucket solo a ID di account specifici

Consigliamo i seguenti strumenti di gestione degli accessi per questo caso d'uso:

- Bucket policy: con le bucket policy, gestisci una policy per ogni bucket. Per ulteriori informazioni, consulta Politiche Bucket per Amazon S3.
- Punti di accesso: i punti di accesso sono denominati endpoint di rete collegati a un bucket. Un bucket può avere fino a 10.000 punti di accesso collegati e per ogni punto di accesso puoi applicare autorizzazioni e controlli di rete distinti per avere un controllo dettagliato sull'accesso ai tuoi oggetti S3. Per ulteriori informazioni, consulta Gestione dell'accesso ai dati con Punti di accesso Amazon S3.

Il Account AWS proprietario o il proprietario del bucket desidera endpoint distinti per ogni utente o applicazione che accede ai propri dati

Consigliamo il seguente strumento di gestione degli accessi per questo caso d'uso:

- Punti di accesso: i punti di accesso sono denominati endpoint di rete collegati a un bucket. Un bucket può avere fino a 10.000 punti di accesso collegati e per ogni punto di accesso puoi applicare autorizzazioni e controlli di rete distinti per avere un controllo dettagliato sull'accesso ai tuoi oggetti S3. Ogni access point applica una policy di access point personalizzata che funziona in combinazione con la policy di bucket collegata al bucket sottostante. Per ulteriori informazioni, consulta Gestione dell'accesso ai dati con Punti di accesso Amazon S3.

Il Account AWS proprietario o il proprietario del bucket deve gestire l'accesso dagli endpoint Virtual Private Cloud (VPC) per S3

Gli endpoint Virtual Private Cloud (VPC) per Amazon S3 sono entità logiche all'interno di un VPC che consentono la connettività solo a S3. Consigliamo i seguenti strumenti di gestione degli accessi per questo caso d'uso:

- Bucket in un'impostazione VPC: puoi utilizzare una policy sui bucket per controllare chi è autorizzato ad accedere ai tuoi bucket e a quali endpoint VPC possono accedere. Per ulteriori informazioni, consulta Controllo dell'accesso dagli endpoint VPC con policy di bucket.
- Punti di accesso: se scegli di configurare i punti di accesso, puoi utilizzare una politica dei punti di accesso. Puoi configurare qualsiasi access point per accettare le richieste solo da un cloud privato virtuale (VPC), in modo da limitare l'accesso ai dati Amazon S3 a una rete privata. È inoltre possibile configurare le impostazioni di blocco dell'accesso pubblico personalizzate per ciascun access point. Per ulteriori informazioni, consulta Gestione dell'accesso ai dati con Punti di accesso Amazon S3.

Il Account AWS proprietario o il proprietario del bucket deve rendere disponibile al pubblico un sito Web statico

Con S3, puoi ospitare un sito Web statico e consentire a chiunque di visualizzarne il contenuto, che è ospitato da un bucket S3.

Consigliamo i seguenti strumenti di gestione degli accessi per questo caso d'uso:

- Amazon CloudFront: questa soluzione consente di ospitare un sito Web statico Amazon S3 al pubblico, continuando allo stesso tempo a bloccare tutti gli accessi pubblici ai contenuti di un bucket. Se desideri mantenere abilitate tutte e quattro le impostazioni di S3 Block Public Access e ospitare un sito Web statico S3, puoi utilizzare Amazon CloudFront Origin Access Control (OAC). Amazon CloudFront offre le funzionalità necessarie per configurare un sito Web statico sicuro. Inoltre, i siti Web statici di Amazon S3 che non utilizzano questa soluzione possono supportare solo endpoint HTTP. CloudFront utilizza lo storage durevole di Amazon S3 fornendo al contempo intestazioni di sicurezza aggiuntive, come HTTPS. HTTPS aggiunge sicurezza crittografando una normale richiesta HTTP e proteggendo contro o più comuni attacchi informatici.

  Per ulteriori informazioni, consulta la sezione Guida introduttiva a un sito Web statico sicuro nella Amazon CloudFront Developer Guide.

- Rendere il tuo bucket Amazon S3 accessibile al pubblico: puoi configurare un bucket da utilizzare come sito Web statico accessibile pubblicamente.

  > **Warning**
  >
  > Non consigliamo questo metodo. Ti consigliamo invece di utilizzare siti Web statici di Amazon S3 come parte di Amazon. CloudFront Per ulteriori informazioni, consulta l'opzione precedente o la Guida introduttiva a un sito Web statico sicuro.

  Per creare un sito Web statico Amazon S3, senza Amazon CloudFront, devi innanzitutto disattivare tutte le impostazioni di Block Public Access. Durante la scrittura della policy del bucket per il sito Web statico, assicurati di consentire solo operazioni `s3:GetObject`, non autorizzazioni `ListObject` o `PutObject`. Questo aiuta a garantire che gli utenti non possano visualizzare tutti gli oggetti nel bucket o aggiungere i propri contenuti. Per ulteriori informazioni, consulta Impostazione delle autorizzazioni per l'accesso al sito Web.

Il Account AWS proprietario o il proprietario del bucket desidera rendere il contenuto di un bucket disponibile al pubblico

Quando si crea un nuovo bucket Amazon S3, l'impostazione Block Public Access è abilitata per impostazione predefinita. Per ulteriori informazioni sul blocco dell'accesso pubblico, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

Non è consigliabile consentire l'accesso pubblico al tuo bucket. Tuttavia, se è necessario farlo per un caso d'uso particolare, consigliamo il seguente strumento di gestione degli accessi per questo caso d'uso:

- Disattiva l'impostazione Block Public Access: il proprietario di un bucket può consentire richieste non autenticate al bucket. Ad esempio, le richieste di oggetti PUT non autenticate sono consentite quando un bucket ha una politica di bucket pubblica o quando un bucket ACL concede l'accesso pubblico. Tutte le richieste non autenticate vengono effettuate da altri utenti arbitrari AWS o addirittura da utenti anonimi non autenticati. Questo utente viene rappresentato dall'ID utente canonico specifico `65a011a29cdf8ec533ec3d1ccaae921c` nelle ACL. Se un oggetto viene caricato su un `WRITE` `oFULL_CONTROL`, ciò concede specificamente l'accesso al gruppo Tutti gli utenti o all'utente anonimo. Per ulteriori informazioni sulle policy dei bucket pubbliche e sulle liste di controllo accessi (ACL) pubbliche, consulta Significato di "pubblico".

Il Account AWS proprietario o il proprietario del bucket ha superato i limiti di dimensione della politica di accesso

Sia le politiche dei bucket che le politiche basate sull'identità hanno un limite di dimensione di 20 KB. Se i requisiti di autorizzazione di accesso sono complessi, potresti superare questo limite di dimensione.

Abbiamo consigliato i seguenti strumenti di gestione degli accessi per questo caso d'uso:

- Punti di accesso: utilizza i punti di accesso se questo è adatto al tuo caso d'uso. Per quanto riguarda i punti di accesso, ogni bucket ha più endpoint di rete denominati, ciascuno con una propria politica di punto di accesso che funziona con la policy del bucket sottostante. Tuttavia, gli access point possono agire solo sugli oggetti, non sui bucket, e non supportano la replica tra regioni. Per ulteriori informazioni, consulta Gestione dell'accesso ai dati con Punti di accesso Amazon S3.
- S3 Access Grants: utilizza S3 Access Grants, che supporta un numero molto elevato di concessioni che danno accesso a bucket, prefissi o oggetti. Per ulteriori informazioni, consulta Nozioni di base su S3 Access Grants.

Il ruolo di Account AWS proprietario o amministratore desidera concedere l'accesso a bucket, prefisso o oggetto direttamente a utenti o gruppi in una directory aziendale

Invece di gestire utenti, gruppi e ruoli tramite AWS Identity and Access Management (IAM), puoi aggiungere la tua directory aziendale a. AWS IAM Identity Center Per ulteriori informazioni, consulta Cos'è IAM Identity Center? .

Dopo aver aggiunto la rubrica aziendale AWS IAM Identity Center, ti consigliamo di utilizzare il seguente strumento di gestione degli accessi per concedere alle identità della directory aziendale l'accesso alle tue risorse S3:

- S3 Access Grants: utilizza S3 Access Grants, che supporta la concessione dell'accesso a utenti o ruoli nella directory aziendale. Per ulteriori informazioni, consulta Nozioni di base su S3 Access Grants.

Il Account AWS proprietario o il proprietario del bucket desidera consentire al AWS CloudFront servizio di accedere alla scrittura di log su un bucket S3 CloudFront

Abbiamo consigliato il seguente strumento di gestione degli accessi per questo caso d'uso:

- Bucket ACL: l'unico caso d'uso consigliato per gli ACL dei bucket è concedere autorizzazioni a determinati utenti, Servizi AWS come l'account Amazon. CloudFront `awslogsdelivery` Quando crei o aggiorni una distribuzione e attivi la CloudFront registrazione, CloudFront aggiorna l'ACL del bucket per concedere all'`awslogsdelivery`account `FULL_CONTROL` le autorizzazioni per scrivere i log nel bucket. Per ulteriori informazioni, consulta la sezione Autorizzazioni necessarie per configurare la registrazione standard e accedere ai file di registro nella Amazon CloudFront Developer Guide. Se il bucket che memorizza i log utilizza l'impostazione imposta dal proprietario del bucket per S3 Object Ownership per disattivare gli ACL, non può scrivere log nel bucket. CloudFront Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

In qualità di proprietario del bucket, desideri mantenere il pieno controllo degli oggetti aggiunti al bucket da altri utenti

Puoi concedere ad altri account l'accesso per caricare oggetti nel tuo bucket utilizzando una policy sui bucket, un punto di accesso o S3 Access Grants. Se hai concesso l'accesso multiaccount al tuo bucket, puoi assicurarti che tutti gli oggetti caricati nel tuo bucket rimangano sotto il tuo pieno controllo.

Abbiamo consigliato il seguente strumento di gestione degli accessi per questo caso d'uso:

- Proprietà dell'oggetto: mantiene l'impostazione a livello di bucket Object Ownership sull'impostazione predefinita del proprietario del bucket.

Risoluzione dei problemi di gestione dell'accesso

Le seguenti risorse possono aiutarti a risolvere eventuali problemi relativi alla gestione degli accessi S3:

Risoluzione dei problemi relativi agli errori di accesso negato (403 Accesso negato)

Se riscontri problemi di negazione dell'accesso, controlla le impostazioni a livello di account e a livello di bucket. Inoltre, controlla la funzionalità di gestione degli accessi che stai utilizzando per concedere l'accesso per assicurarti che la politica, l'impostazione o la configurazione siano corrette. Per ulteriori informazioni sulle cause più comuni degli errori di accesso negato (403 Accesso negato) in Amazon S3, consulta Risoluzione dei problemi relativi agli errori di accesso negato (403 Accesso negato) in Amazon S3.

IAM Access Analyzer per S3

Se non desideri rendere disponibile al pubblico nessuna delle tue risorse o se desideri limitare l'accesso pubblico alle tue risorse, puoi utilizzare IAM Access Analyzer per S3. Sulla console Amazon S3, usa IAM Access Analyzer per S3 per esaminare tutti i bucket che dispongono di elenchi di controllo degli accessi ai bucket (ACL), policy dei bucket o policy dei punti di accesso che garantiscono l'accesso pubblico o condiviso. IAM Access Analyzer for S3 ti avvisa dei bucket configurati per consentire l'accesso a chiunque su Internet o altro, anche all'esterno della tua organizzazione. Account AWS Account AWS Per ogni bucket pubblico o condiviso, vengono visualizzati risultati che riportano l'origine e il livello di accesso pubblico o condiviso.

In IAM Access Analyzer for S3, puoi bloccare tutti gli accessi pubblici a un bucket con una sola azione. Ti consigliamo di bloccare tutti gli accessi pubblici ai tuoi bucket, a meno che tu non richieda l'accesso pubblico per supportare un caso d'uso specifico. Prima di bloccare tutti gli accessi pubblici, assicurati che le tue applicazioni continuino a funzionare correttamente anche senza accesso pubblico. Per ulteriori informazioni, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

Puoi anche rivedere le impostazioni di autorizzazione a livello di bucket per configurare livelli di accesso dettagliati. Per casi d'uso specifici e verificati che richiedono l'accesso pubblico o condiviso, puoi confermare e registrare l'intenzione del bucket di rimanere pubblico o condiviso archiviando i risultati per il bucket. Puoi consultare e modificare le configurazioni relative al bucket in qualsiasi momento. Inoltre, puoi scaricare i risultati come report CSV per scopi di audit.

IAM Access Analyzer per S3 è disponibile senza costi aggiuntivi nella console di Amazon S3. IAM Access Analyzer per S3 è basato su AWS Identity and Access Management (IAM) IAM Access Analyzer. Per utilizzare IAM Access Analyzer for S3 sulla console Amazon S3, devi visitare la console IAM e creare un analizzatore a livello di account in IAM Access Analyzer per ogni singola regione.

Per ulteriori informazioni su IAM Access Analyzer per S3, consultare Revisione dell'accesso al bucket tramite IAM Access Analyzer per S3.

Registrazione di log e monitoraggio

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni delle soluzioni Amazon S3 in modo da poter eseguire più facilmente il debug di un errore di accesso. La registrazione può fornire informazioni sugli errori ricevuti dagli utenti e su quando e quali richieste vengono effettuate. AWS fornisce diversi strumenti per il monitoraggio delle risorse Amazon S3, come i seguenti:

- AWS CloudTrail
- Log di accesso Amazon S3

- AWS Trusted Advisor
- Amazon CloudWatch

Per ulteriori informazioni, consulta Registrazione e monitoraggio in Amazon S3.

Identity and Access Management per Amazon S3

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM controllano chi può essere autenticato (effettuare l'accesso) e autorizzato (disporre delle autorizzazioni) per utilizzare le risorse Amazon S3. IAM è uno strumento Servizio AWS che puoi utilizzare senza costi aggiuntivi.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Argomenti

- Destinatari
- Autenticazione con identità
- Gestione dell'accesso con policy
- Come funziona Amazon S3 con IAM
- Politiche e autorizzazioni in Amazon S3
- Politiche Bucket per Amazon S3
- Policy basate sull'identità per Amazon S3
- Procedure dettagliate che utilizzano policy per gestire l'accesso alle risorse Amazon S3
- In che modo Amazon S3 autorizza una richiesta
- AWS politiche gestite per Amazon S3
- Utilizzo dei ruoli collegati ai servizi per Amazon S3 Storage Lens
- Risoluzione dei problemi di identità e accesso ad Amazon S3

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in Amazon S3.

Utente del servizio: se utilizzi il servizio Amazon S3 per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più funzionalità di Amazon S3 per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità in Amazon S3, consulta. Risoluzione dei problemi di identità e accesso ad Amazon S3

Amministratore del servizio: se sei responsabile delle risorse Amazon S3 della tua azienda, probabilmente hai pieno accesso ad Amazon S3. È tuo compito determinare a quali funzionalità e risorse di Amazon S3 devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM con Amazon S3, consulta. Come funziona Amazon S3 con IAM

Amministratore IAM: se sei un amministratore IAM, potresti voler conoscere i dettagli su come scrivere policy per gestire l'accesso ad Amazon S3. Per visualizzare esempi di policy basate sull'identità di Amazon S3 che puoi utilizzare in IAM, consulta. Policy basate sull'identità per Amazon S3

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta Signing AWS API request nella IAM User Guide.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'utente di AWS IAM Identity Center e Utilizzo dell'autenticazione a più fattori (MFA) in AWSnella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione Attività che richiedono le credenziali dell'utente root nella Guida per l'utente IAM.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di

utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni su IAM Identity Center, consulta Cos'è IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, se si hanno casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente IAM.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile avere un gruppo denominato IAMAdmins e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Quando creare un utente IAM (invece di un ruolo) nella Guida per l'utente IAM.

Ruoli IAM

Un ruolo IAM è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Puoi assumere temporaneamente un ruolo IAM in AWS Management Console cambiando ruolo. Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per

ulteriori informazioni sulla federazione dei ruoli, consulta Creazione di un ruolo per un provider di identità di terza parte nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .

- Autorizzazioni utente IAM temporanee: un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per conoscere la differenza tra ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta Cross Account Resource Access in IAM nella IAM User Guide.
- Accesso tra servizi: alcuni Servizi AWS utilizzano funzionalità in altri. Servizi AWS Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - Sessioni di accesso diretto (FAS): quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.
  - Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire azioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWSnella Guida per l'utente IAM.
  - Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del

servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che AWS CLI effettuano richieste API. AWS Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un AWS ruolo a un'istanza EC2 e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2 nella Guida per l'utente IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta Quando creare un ruolo IAM (invece di un utente) nella Guida per l'utente IAM.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'o dall' AWS API.

Policy basate su identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scelta fra policy gestite e policy inline nella Guida per l'utente IAM.

Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Liste di controllo degli accessi (ACL)

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e VPC sono esempi di servizi che supportano gli ACL. AWS WAF Per maggiori informazioni sulle ACL, consulta Panoramica delle liste di controllo degli accessi (ACL) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite delle autorizzazioni è una funzionalità avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal`sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente IAM.
- Politiche di controllo dei servizi (SCP): le SCP sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna. Utente root dell'account AWS Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle policy nella IAM User Guide.

Come funziona Amazon S3 con IAM

Prima di utilizzare IAM per gestire l'accesso ad Amazon S3, scopri quali funzionalità IAM sono disponibili per l'uso con Amazon S3.

Funzionalità IAM che puoi utilizzare con Amazon S3

Funzionalità IAM	Supporto Amazon S3
Policy basate su identità	Sì
Policy basate su risorse	Sì
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione della policy (specifica del servizio)	Sì
Liste di controllo degli accessi (ACL)	Sì
ABAC (tag nelle policy)	Parziale
Credenziali temporanee	Sì
Inoltro delle sessioni di accesso (FAS)	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	Parziale

Per avere una visione di alto livello di come Amazon S3 e AWS altri servizi funzionano con la maggior parte delle funzionalità IAM, AWS consulta i servizi che funzionano con IAM nella IAM User Guide.

Policy basate sull'identità per Amazon S3

Supporta le policy basate su identità	Sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Esempi di policy basate sull'identità per Amazon S3

Per visualizzare esempi di policy basate sull'identità di Amazon S3, consulta. Policy basate sull'identità per Amazon S3

Policy basate sulle risorse all'interno di Amazon S3

Supporta le policy basate su risorse	Sì

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta Cross Account Resource Access in IAM nella IAM User Guide.

Il servizio Amazon S3 supporta policy relative ai bucket, policy relative ai punti di accesso e concessioni di accesso:

- Le policy bucket sono politiche basate sulle risorse collegate a un bucket Amazon S3. Una policy sui bucket definisce quali principali attori possono eseguire azioni sul bucket.
- Le policy relative ai punti di accesso sono politiche basate sulle risorse che vengono valutate insieme alla policy bucket sottostante.
- Le concessioni di accesso sono un modello semplificato per definire le autorizzazioni di accesso ai dati in Amazon S3 per prefisso, bucket o oggetto. Per informazioni su S3 Access Grants, consulta. Gestione dell'accesso con S3 Access Grants

Principi per le policy relative ai bucket

L'`Principal`elemento specifica l'utente, l'account, il servizio o altra entità a cui è consentito o negato l'accesso a una risorsa. Di seguito vengono illustrati alcuni esempi di specifica del `Principal`. Per ulteriori informazioni, consulta Principali nella Guida per l'utente di IAM.

Concedi le autorizzazioni a un Account AWS

Per concedere le autorizzazioni a un utente Account AWS, identifica l'account utilizzando il seguente formato.

```
"AWS":"account-ARN"
```

Di seguito vengono mostrati gli esempi.

```
"Principal":{"AWS":"arn:aws:iam::AccountIDWithoutHyphens:root"}
```

```
"Principal":{"AWS":
["arn:aws:iam::AccountID1WithoutHyphens:root","arn:aws:iam::AccountID2WithoutHyphens:root"]}
```

Concessione delle autorizzazioni a un utente IAM

Per concedere l'autorizzazione a un utente IAM nel tuo account, devi fornire una coppia nome-valore `"AWS":"user-ARN"`.

```
"Principal":{"AWS":"arn:aws:iam::account-number-without-hyphens:user/username"}
```

Per esempi dettagliati che forniscono step-by-step istruzioni, vedere Esempio 1: il proprietario del bucket concede agli utenti le autorizzazioni per il bucket eEsempio 3: il proprietario del bucket concede autorizzazioni per gli oggetti che non sono di sua proprietà.

> **Note**
>
> Se un'identità IAM viene eliminata dopo aver aggiornato la policy del bucket, la policy del bucket mostrerà un identificatore univoco nell'elemento principale anziché un ARN. Questi ID univoci non vengono mai riutilizzati, quindi puoi rimuovere in sicurezza i principali con identificatori univoci da tutte le dichiarazioni di policy. Per ulteriori informazioni sugli identificatori unici, consulta Identificatori IAM nella Guida per l'utente di IAM.

Concessione di autorizzazioni anonime

> **Warning**
>
> Si deve prestare attenzione a concedere l'accesso anonimo al proprio bucket Amazon S3. Quando si concede l'accesso anonimo, si consente a qualsiasi persona al mondo di accedere al bucket. È consigliabile non concedere mai alcun tipo di accesso anonimo in scrittura al bucket S3.

Per assegnare l'autorizzazione a tutti, vale a dire l'accesso anonimo, è necessario impostare i caratteri jolly (`"*"`) come valore `Principal`. Ad esempio, se si configura un bucket come un sito Web, si vuole che tutti gli oggetti presenti nel bucket siano pubblicamente accessibili.

```
"Principal":"*"
```

```
"Principal":{"AWS":"*"}
```

L'utilizzo `"Principal": "*"` con `Allow` effetto in una politica basata sulle risorse consente a chiunque, anche se non ha effettuato l'accesso AWS, di accedere alla risorsa.

L'utilizzo di `"Principal" : { "AWS" : "*" }` con un effetto `Allow` in una policy basata sulle risorse consente a qualsiasi utente root, utente IAM, sessione del ruolo assunto o utente federato in qualsiasi account nella stessa partizione di accedere alla tua risorsa.

Per gli utenti anonimi, questi due metodi sono equivalenti. Per ulteriori informazioni, consulta Tutti i principali nella Guida per l'utente di IAM.

Non è possibile utilizzare un carattere jolly per associare parte di un nome di un principale o di un ARN.

> **Important**
>
> Poiché chiunque può crearne una Account AWS, il livello di sicurezza di questi due metodi è equivalente, anche se funzionano in modo diverso.

Limitazione delle autorizzazioni per le risorse

Puoi anche utilizzare la policy delle risorse per limitare l'accesso a risorse che altrimenti sarebbero disponibili per i principali IAM. Usa un'istruzione `Deny` per impedire l'accesso.

L'esempio seguente blocca l'accesso se non viene utilizzato un protocollo di trasporto sicuro:

```
{"Effect": "Deny",
 "Principal": "*",
 "Action": "s3:*",
 "Resource": <bucket ARN>,
 "Condition": {
   "Boolean": { "aws:SecureTransport" : "false"}
 }
}
```

Utilizzare il `"Principal": "*"` in modo che questa restrizione si applichi a tutti è una best practice, anziché tentare di negare l'accesso solo a account o principali specifici utilizzando questo metodo.

Richiedi l'accesso tramite CloudFront URL

Puoi richiedere che i tuoi utenti accedano ai tuoi contenuti Amazon S3 solo utilizzando gli CloudFront URL anziché gli URL di Amazon S3. A tale scopo, crea un CloudFront Origin Access Control (OAC). Quindi, modifica le autorizzazioni sui tuoi dati S3. Nella tua policy sui bucket, puoi impostarla come Principal CloudFront come segue:

```
"Principal":{"Service":"cloudfront.amazonaws.com"}
```

Utilizza un `Condition` elemento della policy per consentire l'accesso CloudFront al bucket solo quando la richiesta è per conto della CloudFront distribuzione che contiene l'origine S3.

```
        "Condition": {
```

```
          "StringEquals": {
             "AWS:SourceArn":
"arn:aws:cloudfront::111122223333:distribution/CloudFront-distribution-ID"
          }
       }
```

Per ulteriori informazioni su come richiedere l'accesso a S3 tramite CloudFront URL, consulta Limitazione dell'accesso a un'origine Amazon Simple Storage Service nella Amazon CloudFront Developer Guide. Per ulteriori informazioni sui vantaggi in termini di sicurezza e privacy derivanti dall'utilizzo di Amazon CloudFront, consulta Configurazione dell'accesso sicuro e limitazione dell'accesso ai contenuti.

Esempi di policy basate sulle risorse per Amazon S3

- Per visualizzare esempi di policy per i bucket Amazon S3, consulta. Politiche Bucket per Amazon S3
- Per visualizzare esempi di policy per i punti di accesso, consulta. Configurazione delle policy IAM per l'utilizzo degli access point

Azioni politiche per Amazon S3

Supporta le operazioni di policy	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Action`di una policy JSON descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le azioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Di seguito vengono illustrati i diversi tipi di relazione di mappatura tra le operazioni dell'API S3 e le azioni politiche richieste.

- O ne-to-one mappatura con lo stesso nome. Ad esempio, per utilizzare l'operazione `PutBucketPolicy` API, è necessaria l'azione `s3:PutBucketPolicy` politica.
- O ne-to-one mappatura con nomi diversi. Ad esempio, per utilizzare l'operazione `ListObjectsV2` API, è necessaria l'azione `s3:ListBucket` politica.
- ne-to-many Mappatura O. Ad esempio, per utilizzare l'operazione `HeadObject` API, `s3:GetObject` è necessario. Inoltre, quando utilizzi S3 Object Lock e desideri ottenere lo stato Legal Hold o le impostazioni di conservazione di un oggetto, sono necessarie anche le azioni corrispondenti `s3:GetObjectLegalHold` o `s3:GetObjectRetention` politiche prima di poter utilizzare l'operazione `HeadObject` API.
- any-to-one Mappatura M. Ad esempio, per utilizzare le operazioni `ListObjectsV2` o `HeadBucket` API, è richiesta l'azione `s3:ListBucket` politica.

Per visualizzare un elenco delle azioni di Amazon S3 da utilizzare nelle politiche, consulta Azioni definite da Amazon S3 nel Service Authorization Reference. Per un elenco completo delle operazioni API di Amazon S3, consulta Amazon S3 API Actions nel riferimento alle API di Amazon Simple Storage Service.

Le azioni politiche in Amazon S3 utilizzano il seguente prefisso prima dell'azione:

```
s3
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [
      "s3:action1",
      "s3:action2"
         ]
```

Operazioni relative ai bucket

Le operazioni bucket sono operazioni API S3 che operano sul tipo di risorsa bucket. For example: `CreateBucket`, `ListObjectsV2` e `PutBucketPolicy`. Le azioni delle policy di S3 per le operazioni con i bucket richiedono che l'`Resource`elemento nelle bucket policy o nelle policy basate sull'identità IAM sia l'identificatore Amazon Resource Name (ARN) del tipo di bucket S3 nel seguente formato di esempio.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
```

La seguente policy sui bucket concede all'utente *Akua* con account *12345678901* l'autorizzazione a eseguire l'operazione API V2 e ad elencare gli oggetti in un bucket S3. `s3:ListBucket` ListObjects

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Akua to list objects in the bucket",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": [
            "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
        }
    ]
}
```

Operazioni bucket nelle politiche dei punti di accesso

Le autorizzazioni concesse in una policy sui punti di accesso sono efficaci solo se il bucket sottostante consente le stesse autorizzazioni. Quando utilizzi S3 Access Points, devi delegare il controllo degli accessi dal bucket al punto di accesso o aggiungere le stesse autorizzazioni nelle policy dei punti di accesso alla policy del bucket sottostante. Per ulteriori informazioni, consulta Configurazione delle policy IAM per l'utilizzo degli access point. Nelle policy dei punti di accesso, le azioni delle policy di S3 per le operazioni con i bucket richiedono l'utilizzo dell'`accesspoint`ARN per l'`Resource`elemento nel seguente formato.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-ACCESS-POINT"
```

La seguente politica del punto di accesso concede all'utente *Akua* con account *12345678901* l'`s3:ListBucket`autorizzazione a eseguire l'operazione API ListObjectsV2 tramite il punto di accesso S3 *DOC-EXAMPLE-ACCESS-POINT* per elencare gli oggetti nel bucket associato al punto di accesso.

```
{
```

```
    "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**
>
> Non tutte le operazioni relative ai bucket sono supportate da S3 Access Point. Per ulteriori informazioni, consulta Compatibilità dei punti di accesso con le operazioni S3.

Operazioni sugli oggetti

Le operazioni sugli oggetti sono operazioni dell'API S3 che agiscono sul tipo di risorsa dell'oggetto. For example: `GetObject`, `PutObject` e `DeleteObject`. Le azioni delle policy di S3 per le operazioni sugli oggetti richiedono che l'`Resource`elemento nelle policy sia l'ARN dell'oggetto S3 nei seguenti formati di esempio.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
```

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/prefix/*"
```

> **Note**
>
> L'ARN dell'oggetto deve contenere una barra dopo il nome del bucket, come illustrato negli esempi precedenti.

La seguente policy sui bucket concede all'utente *Akua* con account *12345678901* l'`s3:PutObject`autorizzazione a eseguire l'operazione API per caricare oggetti in un bucket S3. PutObject

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Akua to upload objects",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": [
            "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        }
    ]
}
```

Operazioni sugli oggetti nelle politiche dei punti di accesso

Quando utilizzi i punti di accesso S3 per controllare l'accesso alle operazioni sugli oggetti, puoi utilizzare le policy dei punti di accesso. Quando si utilizzano le policy dei punti di accesso, le azioni delle policy di S3 per le operazioni sugli oggetti richiedono l'utilizzo dell'`accesspoint`ARN per `Resource` l'elemento nel seguente formato:. `arn:aws:s3:region:account-id:accesspoint/access-point-name/object/resource` Per le operazioni sugli oggetti che utilizzano il punto di accesso, è necessario includere il `/object/` valore dopo l'intero ARN del punto di accesso nell'`Resource`elemento. Ecco alcuni esempi.

```
"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 seguente politica del punto di accesso concede all'utente *Akua* con account *12345678901* l'`s3:GetObject`autorizzazione a eseguire l'operazione GetObjectAPI tramite il punto di accesso *DOC-EXAMPLE-ACCESS-POINT* su tutti gli oggetti nel bucket associato al punto di accesso.

```
{
    "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**
>
> Non tutte le operazioni sugli oggetti sono supportate da S3 Access Point. Per ulteriori informazioni, consulta Compatibilità dei punti di accesso con le operazioni S3.

Operazioni sui punti di accesso

Le operazioni sui punti di accesso sono operazioni dell'API S3 che operano sul tipo di `accesspoint` risorsa. For example: `CreateAccessPoint`, `DeleteAccessPoint` e `GetAccessPointPolicy`. Le azioni policy di S3 per le operazioni sui punti di accesso possono essere utilizzate solo nelle policy basate sull'identità IAM, non nelle policy bucket o nelle policy dei punti di accesso. Le operazioni relative ai punti di accesso richiedono che l'`Resource`elemento sia l'`accesspoint`ARN nel formato di esempio seguente.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-ACCESS-POINT"
```

La seguente policy basata sull'identità IAM concede l's3:GetAccessPointPolicyautorizzazione a eseguire l'operazione GetAccessPointPolicyAPI sul punto di accesso 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"
        }
    ]
}
```

Quando utilizzi gli Access Point, per controllare l'accesso alle operazioni del bucket, vedi; per controllare l'accesso alle operazioni sugli oggetti, vedi. Operazioni bucket nelle politiche dei punti di accesso Operazioni sugli oggetti nelle politiche dei punti di accesso Per ulteriori informazioni su come configurare le politiche dei punti di accesso, vedereConfigurazione delle policy IAM per l'utilizzo degli access point.

Operazioni del punto di accesso Object Lambda

Con Amazon S3 Object Lambda puoi aggiungere codice personalizzato alle richieste `GET`, `LIST` e `HEAD` di Amazon S3 per modificare ed elaborare i dati restituiti a un'applicazione. È possibile effettuare richieste tramite un punto di accesso Object Lambda, che funziona allo stesso modo delle richieste tramite altri punti di accesso. Per ulteriori informazioni, consulta Trasformazione di oggetti con S3 Object Lambda.

Per ulteriori informazioni su come configurare le politiche per le operazioni degli access point Object Lambda, vedere. Configurazione delle policy IAM per i punti di accesso Lambda per oggetti

Operazioni con punti di accesso multiregionali

Un punto di accesso multiregionale fornisce un endpoint globale che le applicazioni possono utilizzare per soddisfare le richieste provenienti da bucket S3 che si trovano in più bucket. Regione AWSÈ possibile utilizzare un punto di accesso multiregionale per creare applicazioni multiregionali con la stessa architettura utilizzata in una singola regione e quindi eseguire tali applicazioni in qualsiasi parte del mondo. Per ulteriori informazioni, consulta Punti di accesso multi-regione in Amazon S3.

Per ulteriori informazioni su come configurare le politiche per le operazioni dei punti di accesso multiregionali, vedere. Esempi di policy dei punti di accesso multi-regione

Operazioni di lavoro in batch

Le operazioni di lavoro (Batch Operations) sono operazioni API S3 che operano sul tipo di risorsa lavoro. Ad esempio `DescribeJob` e `CreateJob`. Le azioni policy di S3 per le job operations possono essere utilizzate solo nelle policy basate sull'identità IAM, non nelle policy bucket. Inoltre, le operazioni di lavoro richiedono che l'`Resource`elemento nelle politiche basate sull'identità IAM sia l'`job`ARN nel seguente formato di esempio.

```
"Resource": "arn:aws:s3:*:123456789012:job/*"
```

La seguente policy basata sull'identità IAM concede l's3:DescribeJobautorizzazione a eseguire l'operazione DescribeJobAPI su S3 Batch Operations Job DOC-EXAMPLE-JOB.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow describing the Batch operation job DOC-EXAMPLE-JOB",
            "Effect": "Allow",
            "Action": [
            "s3:DescribeJob"
            ],
            "Resource": "arn:aws:s3:*:123456789012:job/DOC-EXAMPLE-JOB"
        }
    ]
}
```

Operazioni di configurazione di S3 Storage Lens

Per ulteriori informazioni su come configurare le operazioni di configurazione di S3 Storage Lens, consulta. Autorizzazioni Amazon S3 Storage Lens

Operazioni sull'account

Le operazioni dell'account sono operazioni API S3 che operano a livello di account. Ad esempio, `GetPublicAccessBlock` (per account). L'account non è un tipo di risorsa definito da Amazon

S3. Le azioni delle policy di S3 per le operazioni degli account possono essere utilizzate solo nelle politiche basate sull'identità IAM, non nelle policy bucket. Inoltre, le operazioni relative agli account richiedono che l'`Resource`elemento contenuto nelle policy IAM basate sull'identità sia. `"*"`

La seguente policy basata sull'identità IAM concede l'`s3:GetAccountPublicAccessBlock`autorizzazione a eseguire l'operazione GetPublicAccessBlockAPI a livello di account e recuperare le impostazioni del Public Access Block a livello di account.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"Allow retrieving the account-level Public Access Block settings",
         "Effect":"Allow",
         "Action":[
            "s3:GetAccountPublicAccessBlock"
         ],
         "Resource":[
            "*"
         ]
       }
    ]
}
```

Esempi di policy per Amazon S3

- Per visualizzare esempi di policy basate sull'identità di Amazon S3, consulta. Policy basate sull'identità per Amazon S3
- Per visualizzare esempi di policy basate sulle risorse di Amazon S3, consulta e. Politiche Bucket per Amazon S3 Configurazione delle policy IAM per l'utilizzo degli access point

Risorse relative alle policy per Amazon S3

Supporta le risorse di policy	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource`della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource`o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Alcune azioni dell'API Amazon S3 supportano più risorse. Ad esempio, `s3:GetObject` accede a EXAMPLE-RESOURCE-1 e EXAMPLE-RESOURCE-2, quindi un principale deve disporre delle autorizzazioni per accedere a entrambe le risorse. Per specificare più risorse in una singola istruzione, separa gli ARN con le virgole.

```
"Resource": [
      "EXAMPLE-RESOURCE-1",
      "EXAMPLE-RESOURCE-2"
```

Le risorse in Amazon S3 sono bucket, oggetti, punti di accesso o job. In una policy, utilizza l'Amazon Resource Name (ARN) del bucket, dell'oggetto, del punto di accesso o del job per identificare la risorsa.

Per visualizzare un elenco completo dei tipi di risorse Amazon S3 e dei relativi ARN, consulta Resources defined by Amazon S3 nel Service Authorization Reference. Per sapere con quali azioni è possibile specificare l'ARN di ogni risorsa, consulta Azioni definite da Amazon S3.

Wildcard per gli ARN delle risorse

È possibile utilizzare caratteri jolly come parte dell'ARN della risorsa. È possibile utilizzare caratteri jolly (* e ?) all'interno di un segmento ARN (le parti separate da due punti). Un asterisco (*) rappresenta qualsiasi combinazione di zero o più caratteri, mentre un punto interrogativo (?) rappresenta qualsiasi singolo carattere. È possibile utilizzare * o ? multipli in ogni segmento, ma un carattere jolly non può separare i segmenti.

- Il seguente ARN utilizza il carattere jolly * nella parte relativa all'ID dell'ARN per identificare tutti gli oggetti nel bucket `examplebucket`.

```
arn:aws:s3:::examplebucket/*
```

- Il seguente ARN utilizza * per indicare tutti i bucket e gli oggetti S3.

```
arn:aws:s3:::*
```

- Il seguente ARN utilizza entrambi i caratteri jolly * e ?, nella parte `relative-ID`. Identifica tutti gli oggetti nei bucket quali `example1bucket`, `example2bucket`, `example3bucket` e così via.

```
arn:aws:s3:::example?bucket/*
```

Variabili politiche per gli ARN di risorse

È possibile utilizzare le variabili delle policy negli ARN di Amazon S3. Al momento di valutare la policy, queste variabili predefinite vengono sostituite dai valori corrispondenti. Si supponga di organizzare il bucket come raccolta di cartelle, una cartella per ogni utente. Il nome della cartella corrisponde al nome utente. Per assegnare agli utenti le autorizzazioni per le rispettive cartelle, è possibile specificare la variabile di policy nell'ARN della risorsa:

```
arn:aws:s3:::bucket_name/developers/${aws:username}/
```

In fase di esecuzione, quando la policy viene valutata, la variabile `${aws:username}` nella risorsa ARN viene sostituita dal nome utente della persona che effettua la richiesta.

Esempi di policy per Amazon S3

- Per visualizzare esempi di policy basate sull'identità di Amazon S3, consulta. Policy basate sull'identità per Amazon S3
- Per visualizzare esempi di policy basate sulle risorse di Amazon S3, consulta e. Politiche Bucket per Amazon S3 Configurazione delle policy IAM per l'utilizzo degli access point

Chiavi relative alle condizioni delle politiche per Amazon S3

Supporta le chiavi di condizione delle policy specifiche del servizio	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition`è facoltativo. Puoi compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition`in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND`logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. `OR` Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Ogni chiave di condizione di Amazon S3 è mappata all'intestazione di richiesta con lo stesso nome consentita dall'API su cui è possibile impostare la condizione. Le chiavi di condizione specifiche di Amazon S3 determinano il comportamento delle intestazioni di richiesta con lo stesso nome. Ad esempio, la chiave di condizione `s3:VersionId` utilizzata per concedere l'autorizzazione condizionale definisce il
`s3:GetObjectVersion`
comportamento del parametro di `versionId` query impostato in una richiesta GET Object.

Per visualizzare un elenco di chiavi di condizione di Amazon S3, consulta Chiavi di condizione per Amazon S3 nel Service Authorization Reference. Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, consulta Azioni definite da Amazon S3.

Esempio: limitazione dei caricamenti di oggetti a oggetti con una classe di archiviazione specifica

Supponiamo che l'account A, rappresentato dall'ID account 123456789012, sia proprietario di un bucket. L'amministratore dell'Account A desidera limitare Dave, un utente dell'Account A, in modo che

Dave possa caricare oggetti solo nel bucket archiviato con la classe di archiviazione. `STANDARD_IA` Per limitare il caricamento di oggetti con una classe di storage specifica, l'amministratore dell'Account A può utilizzare la chiave di condizione `s3:x-amz-storage-class`, come illustrato nella policy di bucket di esempio seguente.

```
{
                "Version": "2012-10-17",
                "Statement": [
                  {
                    "Sid": "statement1",
                    "Effect": "Allow",
                    "Principal": {
                      "AWS": "arn:aws:iam::123456789012:user/Dave"
                    },
                    "Action": "s3:PutObject",
                    "Resource": "arn:aws:s3:::example-s3-bucket1/*",
                    "Condition": {
                      "StringEquals": {
                        "s3:x-amz-storage-class": [
                          "STANDARD_IA"
                        ]
                      }
                    }
                  }
                ]
            }
```

Nell'esempio, il blocco `Condition` specifica la condizione `StringEquals` che viene applicata alla coppia chiave-valore `"s3:x-amz-acl":["public-read"]`. Esiste un insieme predefinito di chiavi che possono essere utilizzate nell'espressione di una condizione. L'esempio utilizza la chiave di condizione `s3:x-amz-acl`. Questa condizione prevede che l'utente includa l'intestazione `x-amz-acl` con il valore `public-read` in ogni richiesta PUT object.

Esempi di policy per Amazon S3

- Per visualizzare esempi di policy basate sull'identità di Amazon S3, consulta. Policy basate sull'identità per Amazon S3
- Per visualizzare esempi di policy basate sulle risorse di Amazon S3, consulta e. Politiche Bucket per Amazon S3 Configurazione delle policy IAM per l'utilizzo degli access point

ACL in Amazon S3

Supporta le ACL	Sì

In Amazon S3, le liste di controllo degli accessi (ACL) controllano quali Account AWS sono le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

> **Important**
>
> La maggior parte degli attuali casi d'uso in Amazon S3 non richiede più l'uso delle ACL.

Per informazioni sull'utilizzo degli ACL per controllare l'accesso in Amazon S3, consulta. Gestione degli accessi con le ACL

ABAC con Amazon S3

Supporta ABAC (tag nelle policy)	Parziale

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta Che cos'è ABAC? nella Guida per l'utente IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Per visualizzare esempi di policy basate sull'identità per limitare l'accesso ai job di S3 Batch Operations in base ai tag, consulta. Controllo delle autorizzazioni per S3 Batch Operations utilizzando i tag di processo

ABAC e tag di oggetti

Nelle politiche ABAC, gli oggetti utilizzano `s3:` tag anziché `aws:` tag. Per controllare l'accesso agli oggetti in base ai tag degli oggetti, si forniscono informazioni sui tag nell'elemento condition di una politica utilizzando i seguenti tag:

- `s3:ExistingObjectTag/tag-key`
- `s3:s3:RequestObjectTagKeys`
- `s3:RequestObjectTag/tag-key`

Per informazioni sull'utilizzo dei tag degli oggetti per controllare l'accesso, inclusi esempi di politiche di autorizzazione, vedereTagging e policy di controllo degli accessi.

Utilizzo di credenziali temporanee con Amazon S3

Supporta le credenziali temporanee	Sì

Alcune Servizi AWS non funzionano quando accedi utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla Servizi AWS compatibilità con IAM nella IAM User Guide.

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedi AWS utilizzando il link Single Sign-On (SSO) della tua azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta Cambio di un ruolo (console) nella Guida per l'utente IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare

dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza provvisorie in IAM.

Sessioni di accesso diretto per Amazon S3

Supporta l'inoltro delle sessioni di accesso (FAS)	Sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.

- FAS viene utilizzato da Amazon S3 per effettuare chiamate AWS KMS per decrittografare un oggetto quando SSE-KMS è stato utilizzato per crittografarlo. Per ulteriori informazioni, consulta Utilizzo della crittografia lato server con chiavi (SSE-KMS) AWS KMS.
- S3 Access Grants utilizza anche FAS. Dopo aver creato una concessione di accesso ai dati S3 per una particolare identità, l'assegnatario richiede una credenziale temporanea a S3 Access Grants. S3 Access Grants ottiene una credenziale temporanea per il richiedente e la fornisce al richiedente. AWS STS Per ulteriori informazioni, consulta Richiedi l'accesso ai dati di Amazon S3 tramite S3 Access Grants.

Ruoli di servizio per Amazon S3

Supporta i ruoli di servizio	Sì

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWSnella Guida per l'utente IAM.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe compromettere la funzionalità di Amazon S3. Modifica i ruoli di servizio solo quando Amazon S3 fornisce indicazioni in tal senso.

Ruoli collegati ai servizi per Amazon S3

Supporta i ruoli collegati ai servizi	Parziale

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Amazon S3 supporta i ruoli collegati ai servizi per Amazon S3 Storage Lens. Per dettagli sulla creazione o la gestione di ruoli collegati ai servizi Amazon S3, consulta. Utilizzo dei ruoli collegati ai servizi per Amazon S3 Storage Lens

Servizio Amazon S3 come principale

Nome del servizio nella politica	Funzionalità S3	Ulteriori informazioni
`s3.amazonaws.com`	Replica di Amazon S3	Configurazione della replica in tempo reale
`s3.amazonaws.com`	Notifiche di eventi S3	Notifiche di eventi Amazon S3
`s3.amazonaws.com`	Inventario S3	Amazon S3 Inventory
`access-grants.s3.amazonaws.com`	S3 Access Grants	Registrazione di una posizione
`batchoperations.s3.amazonaws.com`	Operazioni in batch S3	Concessione delle autorizzazioni per le operazioni in batch Amazon S3

Nome del servizio nella politica	Funzionalità S3	Ulteriori informazioni
`logging.s3.amazonaws.com`	Registrazione degli accessi al server S3	Abilitazione della registrazione degli accessi al server Amazon S3
`storage-lens.s3.amazonaws.com`	S3 Storage Lens	Visualizzazione dei parametri di Amazon S3 Storage Lens utilizzando una esportazione di dati

Politiche e autorizzazioni in Amazon S3

In questa pagina viene fornita una panoramica delle policy utente e del bucket in Amazon S3 e vengono descritti gli elementi di base di una policy. Ogni elemento elencato è collegato a ulteriori dettagli ed esempi su come usare l'elemento.

Per un elenco completo di azioni, risorse e condizioni di Amazon S3, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.

In termini basilari, una policy contiene i seguenti elementi:

- Risorsa: il bucket, l'oggetto, il punto di accesso o il processo di Amazon S3 a cui si applica la policy. Usa il nome della risorsa Amazon (ARN) del bucket, dell'oggetto, del punto di accesso o del processo per identificare la risorsa.

  Un esempio di operazioni a livello di bucket:

  - `"Resource": "arn:aws:s3:::bucket_name"`.

  Esempi di operazioni a livello di oggetto:

  - `"Resource": "arn:aws:s3:::bucket_name/*"` per tutti gli oggetti nel bucket.

  - `"Resource": "arn:aws:s3:::bucket_name/prefix/*"` per oggetti con un determinato prefisso nel bucket.

  Per ulteriori informazioni, consulta Risorse relative alle policy per Amazon S3.

- Operazioni - Per ciascuna risorsa, Amazon S3 supporta un set di operazioni. Vengono identificate le operazioni delle risorse che verranno consentite (o rifiutate) utilizzando le parole chiave dell'operazione.

  L'autorizzazione `s3:ListBucket` permette ad esempio all'utente di utilizzare l'operazione GET Bucket (ListObjects) di Amazon S3. Per ulteriori informazioni sull'uso di operazioni con Simple Storage Service (Amazon S3), consulta Azioni politiche per Amazon S3. Per un elenco completo delle operazioni di Amazon S3, consulta Operazioni.
- Effetto – Effetto che si produce quando l'utente richiede una determinata operazione, che può essere un permesso o un rifiuto.

  Se l'accesso a una risorsa non viene esplicitamente autorizzato (consentito), di fatto è implicitamente rifiutato. È anche possibile negare esplicitamente l'accesso a una risorsa. È possibile eseguire questa operazione per accertarsi che un utente non sia in grado di accedere a una risorsa, anche se l'accesso viene concesso da un'altra policy. Per ulteriori informazioni, consulta Elementi delle policy JSON IAM: Effect.
- Principale – Account o utente a cui viene permesso l'accesso alle operazioni e alle risorse nell'istruzione. In una policy di bucket l'entità principale è l'utente, l'account, il servizio o un'altra entità destinataria di questa autorizzazione. Per ulteriori informazioni, consulta Principi per le policy relative ai bucket.
- Condizione – Condizioni che stabiliscono quando una policy viene applicata. Puoi utilizzare chiavi AWS-wide e chiavi specifiche di Amazon S3 per specificare le condizioni in una policy di accesso di Amazon S3. Per ulteriori informazioni, consulta Esempi di policy Bucket che utilizzano chiavi condizionali.

La seguente policy di bucket di esempio mostra l'effetto, il principal, l'operazione e gli elementi di una risorsa. La policy consente ad Akua, a un utente nell'account *`Account-ID`* `s3:GetObjects3:GetBucketLocation`, e le autorizzazioni `s3:ListBucket` Amazon S3 sul bucket. `awsexamplebucket1`

```
{
    "Version": "2012-10-17",
    "Id": "ExamplePolicy01",
    "Statement": [
        {
            "Sid": "ExampleStatement01",
            "Effect": "Allow",
            "Principal": {
```

```
                "AWS": "arn:aws:iam::123456789012:user/Akua"
            },
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::awsexamplebucket1/*",
                "arn:aws:s3:::awsexamplebucket1"
            ]
        }
    ]
}
```

Per informazioni complete sul linguaggio delle policy, consulta Policies and permissions in IAM e IAM JSON policy reference nella IAM User Guide.

Delega delle autorizzazioni

Se una persona Account AWS possiede una risorsa, può concedere tali autorizzazioni a un'altra. Account AWS Tale account a sua volta può delegare le autorizzazioni, o una parte di esse, agli utenti al suo interno. Questa operazione è denominata delega di autorizzazioni. Tuttavia, un account che riceve le autorizzazioni da un altro account non può delegare l'autorizzazione per più account a un altro. Account AWS

Proprietà di bucket e oggetti di Amazon S3

I bucket e gli oggetti sono risorse di Amazon S3. Per impostazione predefinita, solo il proprietario della risorsa è in grado di accedervi. Il proprietario della risorsa si riferisce a chi crea Account AWS la risorsa. Per esempio:

- La Account AWS persona che usi per creare bucket e caricare oggetti possiede tali risorse.
- Se carichi un oggetto utilizzando le credenziali dell'utente o del ruolo AWS Identity and Access Management (IAM), l'oggetto è il Account AWS proprietario dell'oggetto a cui appartiene l'utente o il ruolo.
- Il proprietario di un bucket può concedere autorizzazioni multiaccount a un altro Account AWS (o agli utenti di un altro account) per caricare oggetti. In questo caso, chi carica Account AWS gli oggetti possiede tali oggetti. Il proprietario del bucket non dispone di autorizzazioni sugli oggetti di proprietà di un altro account, ad eccezione dei casi seguenti:

- È il proprietario del bucket a pagare la fattura. Il proprietario del bucket può rifiutare l'accesso agli oggetti nel bucket o eliminarli, indipendentemente dall'utente a cui appartengono.
- Il proprietario del bucket può archiviare gli oggetti nel bucket o ripristinarli, indipendentemente dall'utente a cui appartengono. L'archiviazione fa riferimento alla classe di storage utilizzata per archiviare gli oggetti. Per ulteriori informazioni, consulta Gestione del ciclo di vita dello storage.

Proprietà e autenticazione delle richieste

Tutte le richieste a un bucket possono essere autenticate o non autenticate. Le richieste autenticate devono includere un valore di firma che autentichi il mittente della richiesta, mentre non è necessario per le richieste non autenticate. Per ulteriori informazioni sull'autenticazione delle richieste, consulta Esecuzione di richieste.

Un proprietario di bucket può consentire richieste non autenticate. Ad esempio, PUT Objectle richieste non autenticate sono consentite quando un bucket ha una policy pubblica per i bucket o quando un bucket ACL concede o `FULL_CONTROL` accede specificamente al gruppo `WRITE` o all'utente anonimo. `All Users` Per ulteriori informazioni sulle policy dei bucket pubbliche e sulle liste di controllo accessi (ACL) pubbliche, consulta Significato di "pubblico".

Tutte le richieste non autenticate sono fatte dall'utente anonimo. Questo utente viene rappresentato dall'ID utente canonico specifico `65a011a29cdf8ec533ec3d1ccaae921c` nelle ACL. Se un oggetto viene caricato in un bucket tramite una richiesta non autenticata, la proprietà dell'oggetto è dell'utente anonimo. L'ACL predefinita dell'oggetto garantisce `FULL_CONTROL` all'utente anonimo in quanto proprietario dell'oggetto. Perciò, Amazon S3 consente alle richieste non autenticate di recuperare l'oggetto o di modificarne l'ACL.

Per impedire all'utente anonimo di modificare oggetti, raccomandiamo di non implementare policy di bucket che consentono scritture pubbliche anonime sul bucket e di non utilizzare delle ACL che concedono all'utente anonimo accesso di scrittura al bucket. Puoi applicare questo comportamento consigliato utilizzando il blocco dell'accesso pubblico di Amazon S3.

Per ulteriori informazioni sul blocco dell'accesso pubblico, consulta Blocco dell'accesso pubblico allo storage Amazon S3. Per ulteriori informazioni sulle ACL, consulta Panoramica delle liste di controllo accessi (ACL).

> **Important**
>
> Ti consigliamo di non utilizzare le credenziali dell'utente Account AWS root per effettuare richieste autenticate. Crea invece un ruolo IAM, concedendo a esso l'accesso completo.

Gli utenti con questo ruolo vengono definiti utenti amministratori. È possibile utilizzare le credenziali assegnate al ruolo di amministratore, anziché le credenziali dell'utente Account AWS root, per interagire AWS ed eseguire attività, come creare un bucket, creare utenti e concedere autorizzazioni. Per ulteriori informazioni, consulta le credenziali AWS di sicurezza nella IAM User Guide e le best practice di sicurezza in IAM nella IAM User Guide.

Politiche Bucket per Amazon S3

Una policy di bucket è una policy basata su risorse che puoi utilizzare per concedere autorizzazioni di accesso al bucket Amazon S3 e agli oggetti che contiene. Solo il proprietario del bucket può associare una policy a un bucket. Le autorizzazioni allegate a un bucket si applicano a tutti gli oggetti del bucket di proprietà del proprietario del bucket. Queste autorizzazioni non si applicano agli oggetti di proprietà di altri. Account AWS

S3 Proprietà dell'oggetto è un'impostazione a livello di bucket Amazon S3 che è possibile utilizzare per controllare la proprietà degli oggetti caricati nel bucket e per disabilitare o abilitare le liste di controllo degli accessi (ACL). Per impostazione predefinita, Proprietà dell'oggetto è impostata su Proprietario del bucket applicato e tutte le ACL sono disabilitate. Il proprietario del bucket dispone di tutti gli oggetti nel bucket e gestisce l'accesso ai dati in maniera esclusiva utilizzando policy.

Le policy Bucket utilizzano un linguaggio di policy basato su JSON AWS Identity and Access Management (IAM). Puoi utilizzare policy di bucket per aggiungere o negare autorizzazioni per gli oggetti in un bucket. I criteri di bucket autorizzano o rifiutano le richieste in base agli elementi inclusi nella policy. Questi elementi possono includere richiedente, operazioni S3, risorse e aspetti o condizioni della richiesta (ad esempio, l'indirizzo IP utilizzato per creare la richiesta).

Ad esempio, è possibile creare una policy di bucket che esegue le seguenti operazioni:

- Concedere ad altri account le autorizzazioni multi-account per il caricamento di oggetti nel bucket S3
- Verificare che il proprietario del bucket abbia il pieno controllo degli oggetti caricati

Per ulteriori informazioni, consulta Esempi di policy relative ai bucket di Amazon S3.

> **Important**
>
> Non puoi utilizzare una policy bucket per impedire eliminazioni o transizioni in base a una regola del ciclo di vita S3. Ad esempio, anche se la tua bucket policy nega tutte le azioni per tutti i principali, la configurazione di S3 Lifecycle funziona comunque normalmente.

Negli argomenti di questa sezione vengono forniti esempi e viene illustrato come aggiungere una policy di bucket nella console S3. Per informazioni sulle politiche basate sull'identità, consulta. Policy basate sull'identità per Amazon S3 Per informazioni sul linguaggio delle policy di bucket, consulta Politiche e autorizzazioni in Amazon S3.

Argomenti

- Aggiunta di una policy di bucket utilizzando la console di Amazon S3
- Controllo dell'accesso dagli endpoint VPC con policy di bucket
- Esempi di policy relative ai bucket di Amazon S3
- Esempi di policy Bucket che utilizzano chiavi condizionali

Aggiunta di una policy di bucket utilizzando la console di Amazon S3

Puoi utilizzare il generatore di policy AWS e la console di Amazon S3 per aggiungere una nuova policy di bucket o modificarne una esistente. Una bucket policy è una policy basata sulle risorse (IAM). AWS Identity and Access Management Aggiungi una policy bucket a un bucket per concedere ad altri utenti Account AWS o a utenti IAM le autorizzazioni di accesso per il bucket e gli oggetti in esso contenuti. Le autorizzazioni relative a un oggetto si applicano solo agli oggetti creati dal proprietario del bucket. Per ulteriori informazioni sulle policy del bucket, consulta Identity and Access Management per Amazon S3.

Assicurati di risolvere avvisi di sicurezza, errori, avvisi generali e suggerimenti da AWS Identity and Access Management Access Analyzer prima di salvare la policy. IAM Access Analyzer esegue controlli della policy per convalidarla in rapporto alla sintassi della policy e alle best practice di IAM. Questi controlli generano risultati e forniscono suggerimenti utili per aiutarti a creare policy funzionali e conformi alle best practice per la sicurezza. Per ulteriori informazioni sulla convalida delle policy tramite IAM Access Analyzer, consulta Convalida delle policy di IAM Access Analyzer nella Guida per l'utente di IAM. Per visualizzare un elenco di avvisi, errori e suggerimenti di IAM Access Analyzer, consulta Riferimento ai controlli delle policy IAM Access Analyzer.

Per istruzioni sulla risoluzione degli errori con una policy, consulta Risoluzione dei problemi relativi agli errori di accesso negato (403 Accesso negato) in Amazon S3.

Per creare o modificare una policy di bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckets (Bucket) scegliere il nome del bucket per il quale si desidera creare o modificare una policy di bucket.
4. Scegli la scheda Autorizzazioni.
5. In Bucket Policy (Policy del bucket) scegliere Edit (Modifica). Viene visualizzata la pagina Edit bucket policy (Modifica policy di bucket).
6. Nella pagina Edit bucket policy (Modifica policy di bucket), esegui una delle seguenti operazioni:

   - Per visualizzare esempi di policy di bucket nella Guida per l'utente di Amazon S3, scegli Policy examples (Esempi di policy).
   - Per generare automaticamente una policy o modificare la sintassi JSON nella sezione Policy, scegli Policy generator (Generatore di policy).

   Se scegli Policy generator, il AWS Policy Generator si apre in una nuova finestra.

   a. Nella pagina AWS Policy Generator (Generatore di policy AWS), per 'opzione Select Type of Policy (Seleziona tipo di policy), scegli S3 Bucket Policy (Policy di bucket S3).
   b. Aggiungi un'istruzione inserendo le informazioni nei campi forniti, quindi scegli Aggiungi istruzione. Ripeti questo passaggio per tutte le istruzioni che desideri aggiungere. Per ulteriori informazioni su questi campi, consulta Riferimento agli elementi delle policy IAM JSON nella Guida per l'utente IAM.

      > **Note**
      >
      > Per comodità, la pagina Edit bucket policy (Modifica policy di bucket) mostra il nome della risorsa Amazon (ARN) del bucket corrente sopra il campo di testo Policy. Puoi copiare questo ARN per utilizzarlo nelle istruzioni alla pagina Generatore di policy di AWS .

   c. Dopo aver aggiunto le istruzioni, scegli Genera policy.

d. Copia il testo della policy generata, scegli Chiudi e torna alla pagina Modifica policy del bucket nella console di Amazon S3.

7. Nella casella Policy, modifica la policy esistente o incolla la bucket policy dal AWS Policy Generator. Assicurati di risolvere gli avvisi di sicurezza, gli errori, gli avvisi generali e i suggerimenti prima di salvare la tua policy.

> **Note**
>
> Le policy di bucket sono limitate a una dimensione di 20 KB.

8. (Facoltativo) Scegli Preview external access (Anteprima accesso esterno) nell'angolo in alto a destra per visualizzare in anteprima in che modo la nuova policy influisce sull'accesso pubblico e multi-account alla risorsa. Prima di salvare la policy, puoi verificare se introduce nuovi risultati di IAM Access Analyzer o risolve i risultati esistenti. Se non è presente uno strumento di analisi attivo, scegli Go to Access Analyzer (Passa a strumento analisi accessi) per creare uno strumento di analisi degli account in IAM Access Analyzer. Per ulteriori informazioni, consulta la sezione Anteprima dell'accesso nella Guida per l'utente di IAM.
9. Scegli Save changes (Salva modifiche), che ti riporterà alla pagina Permissions (Autorizzazioni).

Controllo dell'accesso dagli endpoint VPC con policy di bucket

Puoi utilizzare le policy dei bucket di Amazon S3 per controllare l'accesso ai bucket da endpoint specifici del cloud privato virtuale (VPC) o VPC specifici. Questa sezione contiene esempi di policy relative ai bucket che puoi utilizzare per controllare l'accesso ai bucket Amazon S3 dagli endpoint VPC. Per informazioni su come configurare gli endpoint VPC, consulta Endpoint VPC nella Guida per l'utente di VPC.

Un VPC consente di avviare AWS risorse in una rete virtuale definita dall'utente. Un endpoint VPC ti consente di creare una connessione privata tra il tuo VPC e un altro. Servizio AWS Questa connessione privata non richiede l'accesso via Internet, tramite una connessione di rete privata virtuale (VPN), tramite un'istanza NAT o tramite. AWS Direct Connect

Un endpoint VPC per Amazon S3 è un'entità logica all'interno di un cloud privato virtuale che permette di connettersi esclusivamente ad Amazon S3. L'endpoint VPC instrada le richieste ad Amazon S3 e restituisce le risposte al VPC. Gli endpoint VPC cambiano solo la modalità di instradamento delle richieste. I nomi DNS e gli endpoint pubblici Amazon S3 continueranno a funzionare con gli endpoint VPC. Per informazioni importanti sull'utilizzo degli endpoint VPC con

Amazon S3, consulta Endpoints Gateway e Gateway endpoints per Amazon S3 nella VPC User Guide.

Gli endpoint VPC per Amazon S3 offrono due modi per controllare l'accesso ai dati di Amazon S3:

- È possibile controllare le richieste, gli utenti o i gruppi autorizzati tramite un endpoint VPC specifico. Per informazioni su questo tipo di controllo degli accessi, consulta Controllare l'accesso agli endpoint VPC utilizzando le politiche degli endpoint nella Guida per l'utente VPC.
- È possibile controllare i VPC o gli endpoint VPC che hanno accesso ai bucket utilizzando le policy del bucket Amazon S3. Per alcuni esempi di questo tipo di controllo di accesso basato su policy di bucket, consulta i seguenti argomenti sulla limitazione dell'accesso.

Argomenti

> **Important**
>
> Quando applichi le policy dei bucket di Amazon S3 per gli endpoint VPC descritte in questa sezione, potresti bloccare involontariamente l'accesso al bucket. Le autorizzazioni del bucket che hanno lo scopo di limitare l'accesso del bucket a connessioni originate dall'endpoint VPC possono bloccare tutte le connessioni al bucket. Per informazioni su come risolvere questo problema, vedi Come posso correggere la mia policy bucket quando ha un VPC o un ID endpoint VPC errato? nel Knowledge Center.AWS Support

Limitazione dell'accesso a un endpoint VPC specifico

Di seguito è riportato un esempio di policy del bucket Amazon S3 che limita l'accesso a un bucket specifico, `awsexamplebucket1`, solo dall'endpoint VPC con l'ID `vpce-1a2b3c4d`. Se l'endpoint specificato non viene utilizzato, la policy nega tutti gli accessi al bucket. La `aws:SourceVpce` condizione specifica l'endpoint. La `aws:SourceVpce` condizione non richiede un Amazon Resource Name (ARN) per la risorsa endpoint VPC, ma solo l'ID dell'endpoint VPC. Per ulteriori informazioni sull'utilizzo delle condizioni in una policy, consulta Esempi di policy Bucket che utilizzano chiavi condizionali.

Important

- Prima di utilizzare la policy di esempio seguente, sostituire l'ID endpoint VPC con un valore appropriato per il caso d'uso. In caso contrario, non sarà possibile accedere al bucket.
- Questa policy disabilita l'accesso della console al bucket specificato perché le richieste della console non provengono dall'endpoint VPC specificato.

```
{
   "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"
         }
       }
     }
   ]
}
```

Limitazione dell'accesso a un VPC specifico

Puoi creare una policy di bucket che limita l'accesso a uno specifico VPC utilizzando la condizione `aws:SourceVpc`. Ciò è utile se nello stesso VPC sono configurati più endpoint VPC e desideri gestire l'accesso ai bucket Amazon S3 per tutti gli endpoint. Di seguito è riportato un esempio di policy che nega l'accesso a `awsexamplebucket1` e ai relativi oggetti da qualsiasi punto esterno al VPC `vpc-111bbb22`. Se il VPC specificato non viene utilizzato, la policy nega tutti gli accessi al bucket. Questa istruzione non concede l'accesso al bucket. Per concedere l'accesso, devi aggiungere un'`Allow`istruzione separata. La chiave di `vpc-111bbb22` condizione non richiede un ARN per la risorsa VPC, ma solo l'ID VPC.

> **Important**
>
> - Prima di utilizzare la policy di esempio seguente, sostituire l'ID VPC con un valore appropriato per il caso d'uso. In caso contrario, non sarà possibile accedere al bucket.
> - Questa policy disabilita l'accesso della console al bucket specificato perché le richieste della console non provengono dal VPC specificato.

```
{
   "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"
         }
       }
     }
   ]
}
```

Esempi di policy relative ai bucket di Amazon S3

Con le policy di bucket Amazon S3, puoi proteggere l'accesso agli oggetti nei tuoi bucket, in modo che solo gli utenti con le autorizzazioni appropriate possano accedervi. Puoi persino impedire agli utenti autenticati senza le autorizzazioni appropriate di accedere alle tue risorse Amazon S3.

Questa sezione include esempi di casi d'uso tipici per le policy di bucket. Queste policy di esempio utilizzano *example-s3-bucket* come valore di risorsa. Per testare queste policy, sostituisci *user input placeholders* con le tue informazioni (come il nome del bucket).

Per concedere o negare le autorizzazioni per un insieme di oggetti, puoi utilizzare caratteri jolly (*) nei nomi delle risorse Amazon (ARN) e altri valori. Ad esempio, puoi controllare l'accesso a gruppi di

oggetti che iniziano con un prefisso comune o terminano con un'estensione specifica, ad esempio. `.html`

Per ulteriori informazioni sul linguaggio di policy AWS Identity and Access Management (IAM), consultaPolitiche e autorizzazioni in Amazon S3.

> **Note**
>
> Per testare le autorizzazioni utilizzando la console di Amazon S3, dovrai concedere le autorizzazioni aggiuntive richieste dalla console, ovvero `s3:ListAllMyBuckets`, `s3:GetBucketLocation` e `s3:ListBucket`. Per una procedura dettagliata di esempio che concede autorizzazioni a utenti e le testa utilizzando la console, consulta Procedura guidata: controllo dell'accesso a un bucket con policy utente.

Le risorse aggiuntive per la creazione di policy relative ai bucket includono quanto segue:

- Per un elenco delle azioni, delle risorse e delle chiavi di condizione IAM che puoi utilizzare durante la creazione di una bucket policy, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.
- Per istruzioni sulla creazione della policy S3, consulta Aggiunta di una policy di bucket utilizzando la console di Amazon S3.
- Per risolvere gli errori relativi a una policy, consulta Risoluzione dei problemi relativi agli errori di accesso negato (403 Accesso negato) in Amazon S3.

Argomenti

- Concessione dell'autorizzazione di sola lettura a un utente pubblico anonimo
- Richiesta della crittografia
- Gestione dei bucket tramite ACL predefinite
- Gestione dell'accesso agli oggetti con assegnazione di tag agli oggetti
- Gestione dell'accesso agli oggetti utilizzando chiavi di condizione globali
- Gestione dell'accesso in base a indirizzi IP specifici
- Gestione dell'accesso in base a richieste HTTP o HTTPS
- Gestione dell'accesso utente a cartelle specifiche

- Gestione dell'accesso per i log degli accessi
- Gestione dell'accesso a un Amazon CloudFront OAI
- Gestione dell'accesso per Amazon S3 Storage Lens
- Gestione delle autorizzazioni per i report di S3 Inventory, S3 Analytics e S3 Inventory
- Richiesta dell'autenticazione a più fattori (MFA)
- Impedire agli utenti di eliminare oggetti

Concessione dell'autorizzazione di sola lettura a un utente pubblico anonimo

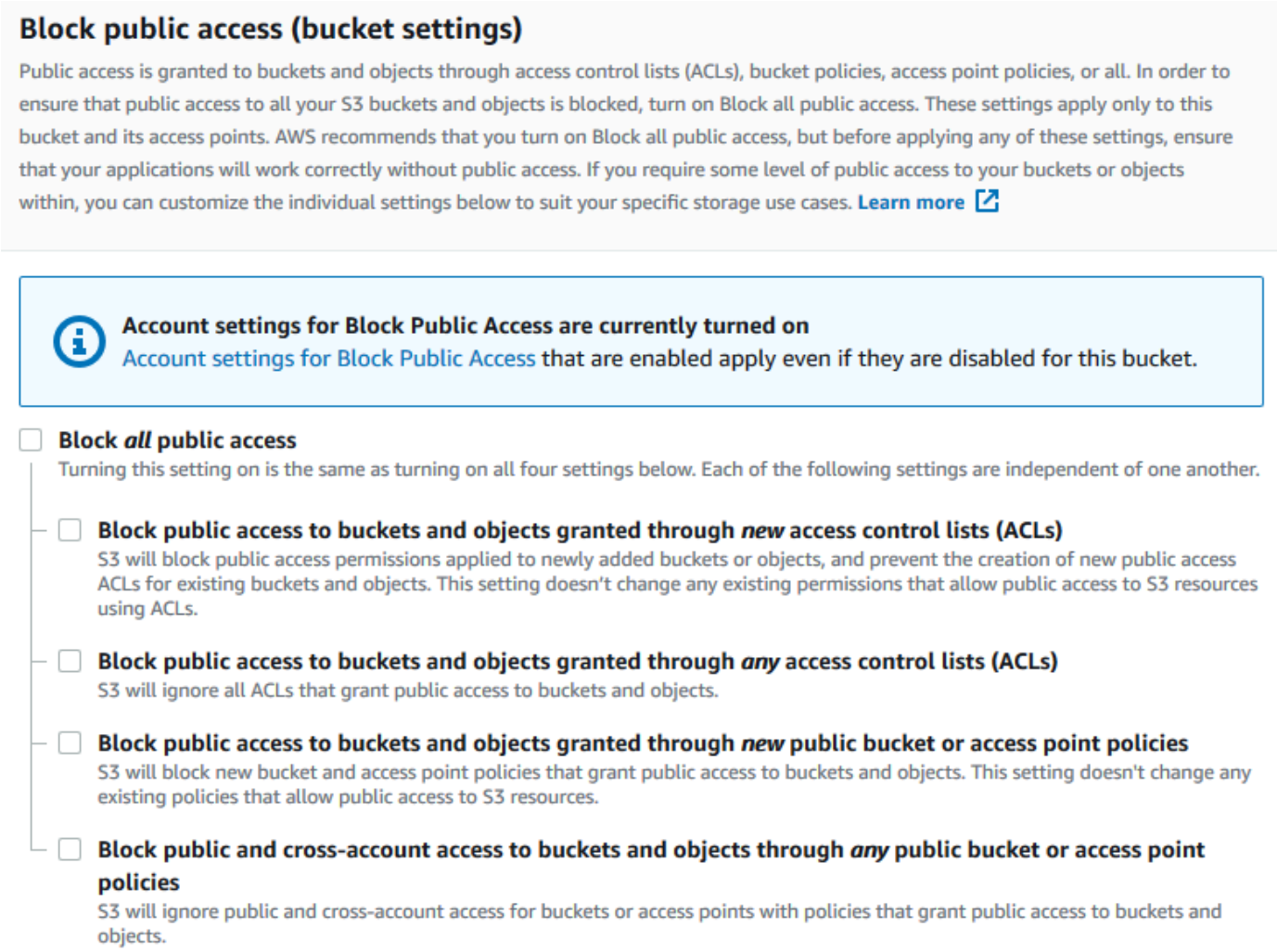
Puoi utilizzare le impostazioni delle policy per concedere l'accesso a utenti anonimi pubblici, il che è utile se stai configurando il tuo bucket come sito web statico. Ciò richiede la disabilitazione del blocco dell'accesso pubblico per il bucket. Per ulteriori informazioni su come eseguire questa operazione e sulla politica richiesta, consulta. Impostazione delle autorizzazioni per l'accesso al sito Web Per informazioni su come configurare politiche più restrittive per lo stesso scopo, vedi Come posso concedere l'accesso pubblico in lettura ad alcuni oggetti nel mio bucket Amazon S3? nel Knowledge Center. AWS

Per impostazione predefinita, Amazon S3 blocca l'accesso pubblico all'account e ai bucket. Per utilizzare un bucket per ospitare un sito Web statico, puoi seguire questa procedura per modificare le impostazioni di blocco dell'accesso pubblico:

> **Warning**
>
> Prima di completare questa fase, consulta Blocco dell'accesso pubblico allo storage Amazon S3 per confermare di avere compreso e accettato i rischi connessi alla concessione di un accesso pubblico. Quando si disattivano le impostazioni di blocco dell'accesso pubblico per rendere pubblico il bucket, chiunque su Internet può accedere al bucket. Consigliamo di bloccare tutti gli accessi pubblici ai bucket.

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
2. Seleziona il nome del bucket configurato come sito Web statico.
3. Seleziona Autorizzazioni.
4. In Blocca accesso pubblico (impostazioni bucket), seleziona Modifica.
5. Deseleziona Blocca tutto l'accesso pubblico, quindi seleziona Salva modifiche.

> **Warning**
>
> Prima di completare questa fase, consulta Blocco dell'accesso pubblico allo storage Amazon S3 per confermare di avere compreso e accettato i rischi connessi alla concessione di un accesso pubblico. Quando si disattivano le impostazioni di blocco dell'accesso pubblico per rendere pubblico il bucket, chiunque su Internet può accedere al bucket. Consigliamo di bloccare tutti gli accessi pubblici ai bucket.



Amazon S3 disattiva le impostazioni di blocco dell'accesso pubblico per il tuo bucket. Per creare un sito web pubblico statico, potrebbe essere necessario modificare anche le impostazioni di blocco dell'accesso pubblico per l'account prima di aggiungere una policy del bucket. Se le

impostazioni dell'account per il blocco dell'accesso pubblico sono attualmente attivate, verrà visualizzata una nota in Blocca accesso pubblico (impostazioni bucket).

Richiesta della crittografia

Richiedi SSE-KMS per tutti gli oggetti scritti in un bucket

La seguente politica di esempio richiede che ogni oggetto scritto nel bucket sia crittografato con la crittografia lato server utilizzando le chiavi AWS Key Management Service (AWS KMS) (SSE-KMS). Se l'oggetto non è crittografato con SSE-KMS, la richiesta viene rifiutata.

```
{
"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"
    }
  }
}]
}
```

Richiedere SSE-KMS con una chiave AWS KMS key specifica per tutti gli oggetti scritti in un bucket

La seguente policy di esempio impedisce la scrittura di qualsiasi oggetto nel bucket se l'oggetto non è crittografato con SSE-KMS mediante un ID chiave KMS specifico. Anche se gli oggetti sono crittografati con SSE-KMS utilizzando un'intestazione per richiesta o una crittografia predefinita per bucket, gli oggetti non possono essere scritti nel bucket se non sono stati crittografati con la chiave KMS specificata. Assicurati di sostituire il nome della risorsa Amazon (ARN) della chiave KMS utilizzata in questo esempio con il nome della risorsa Amazon (ARN) della tua chiave KMS ARN.

```
{
"Version": "2012-10-17",
"Id": "PutObjPolicy",
```

```
"Statement": [{
  "Sid": "DenyObjectsThatAreNotSSEKMSWithSpecificKey",
  "Principal": "*",
  "Effect": "Deny",
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
  "Condition": {
    "ArnNotEqualsIfExists": {
      "s3:x-amz-server-side-encryption-aws-kms-key-id": "arn:aws:kms:us-
east-2:111122223333:key/01234567-89ab-cdef-0123-456789abcdef"
    }
  }
}]
}
```

Gestione dei bucket tramite ACL predefinite

Concessione delle autorizzazioni a più account per caricare oggetti o impostare le ACL degli oggetti per l'accesso pubblico

La politica di esempio seguente concede le autorizzazioni and a più utenti. `s3:PutObject` `s3:PutObjectAcl` Account AWS Inoltre, la politica di esempio richiede che tutte le richieste per queste operazioni includano la lista di controllo degli accessi `public-read` preimpostata (ACL). Per ulteriori informazioni, consulta Azioni politiche per Amazon S3 e Chiavi relative alle condizioni delle politiche per Amazon S3.

> **Warning**
>
> L'ACL `public-read` predefinita consente a chiunque nel mondo di visualizzare gli oggetti nel tuo bucket, indipendentemente dalla sua dislocazione geografica. Procedi con cautela quando concedi l'accesso anonimo al bucket Amazon S3 o disabiliti le impostazioni di blocco dell'accesso pubblico. Quando si concede l'accesso anonimo, si consente a qualsiasi persona al mondo di accedere al bucket. È consigliabile non concedere mai l'accesso anonimo al bucket Amazon S3 a meno che non sia assolutamente necessario, ad esempio con l'hosting di un sito Web statico. Se desideri abilitare le impostazioni di Blocco dell'accesso pubblico Amazon S3 per l'hosting di siti Web statici, consulta Tutorial: Configurazione di un sito Web statico su 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"
                    ]
                }
            }
        }
    ]
}
```

Concedere autorizzazioni multi-account per il caricamento di oggetti a garanzia del controllo completo da parte del proprietario del bucket

L'esempio seguente mostra come consentire a un altro utente Account AWS di caricare oggetti nel tuo bucket, assicurandoti al contempo il pieno controllo degli oggetti caricati. Questa politica concede a uno specifico Account AWS (*111122223333*) la possibilità di caricare oggetti solo se tale account include l'ACL predefinito al `bucket-owner-full-control` momento del caricamento. La condizione `StringEquals` nella policy specifica la chiave di condizione `s3:x-amz-acl` per esprimere il requisito dell'ACL predefinita. Per ulteriori informazioni, consulta Chiavi relative alle condizioni delle politiche per 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"}
      }
    }
  ]
}
```

Gestione dell'accesso agli oggetti con assegnazione di tag agli oggetti

Concedere a un utente autorizzazioni di sola lettura per gli oggetti che hanno una chiave o un valore di tag specifico

La seguente policy di autorizzazione limita un utente a leggere solo gli oggetti con chiave e valore di tag `environment: production`. La policy utilizza la chiave di condizione `s3:ExistingObjectTag` per specificare la chiave e il valore di tag.

```
{
   "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"
            }
         }
      }
   ]
}
```

Limitare le chiavi di tag degli oggetti che gli utenti possono aggiungere

La seguente policy di esempio concede a un utente le autorizzazioni per eseguire l'operazione `s3:PutObjectTagging`, che permette di aggiungere tag a un oggetto esistente. La condizione utilizza la chiave di condizione `s3:RequestObjectTagKeys` per specificare le chiavi di tag consentite, ad esempio `Owner` o `CreationDate`. Per ulteriori informazioni, consulta la sezione Creazione di una condizione con più chiavi o valori nella Guida per l'utente IAM.

La policy garantisce che ogni chiave di tag specificata nella richiesta sia una chiave di tag autorizzata. Il qualificatore `ForAnyValue` nella condizione garantisce che almeno una delle chiavi specificate sia presente nella richiesta.

```
{
   "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"
          ]
        }
      }
    }
  ]
}
```

Richiedere una chiave e un valore di tag specifici per consentire agli utenti di aggiungere tag di oggetti

La seguente policy di esempio concede a un utente le autorizzazioni per eseguire l'operazione `s3:PutObjectTagging`, che permette di aggiungere tag a un oggetto esistente. La condizione

prevede che l'utente includa una chiave di tag specifica (ad esempio, *Project*) con valore impostato su *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"
        }
      }
    }
  ]
}
```

Concedere a un utente di aggiungere solo oggetti che hanno una chiave o un valore di tag specifico

La seguente policy di esempio concede a un utente l'autorizzazione per eseguire l'operazione `s3:PutObject` in modo che possa aggiungere oggetti a un bucket. Tuttavia, l'istruzione `Condition` limita le chiavi e i valori di tag consentiti sugli oggetti caricati. In questo esempio, l'utente può aggiungere al bucket solo oggetti con la chiave di tag specifica (*Department*) con il valore impostato su *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"
            }
        }
    }]
}
```

Gestione dell'accesso agli oggetti utilizzando chiavi di condizione globali

Le chiavi di condizione globali sono chiavi di contesto delle condizioni con un prefisso. `aws` Servizi AWS può supportare chiavi di condizione globali o chiavi specifiche del servizio che includono il prefisso del servizio. È possibile utilizzare l'elemento `Condition` di una policy JSON per confrontare le chiavi in una richiesta con i valori di chiave specificati nella policy.

Limitare l'accesso alle sole consegne dei log degli accessi al server Amazon S3

Nel seguente esempio di bucket policy, la chiave aws:SourceArnglobal condition viene utilizzata per confrontare l'Amazon Resource Name (ARN) della risorsa, effettuando service-to-service una richiesta con l'ARN specificato nella policy. La chiave di condizione globale `aws:SourceArn` viene utilizzata per impedire a un servizio Amazon S3 di essere utilizzato come confused deputy durante le transazioni tra servizi. Solo il servizio Amazon S3 può aggiungere oggetti al bucket Amazon S3.

Questo esempio di policy di bucket concede autorizzazioni `s3:PutObject` al principale del servizio di log (`logging.s3.amazonaws.com`).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutObjectS3ServerAccessLogsPolicy",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET-logs/*",
```

```
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111111111111"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::EXAMPLE-SOURCE-BUCKET"
                }
            }
        },
        {
            "Sid": "RestrictToS3ServerAccessLogs",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET-logs/*",
            "Condition": {
                "ForAllValues:StringNotEquals": {
                    "aws:PrincipalServiceNamesList": "logging.s3.amazonaws.com"
                }
            }
        }
    ]
}
```

Consentire l'accesso solo alla tua organizzazione

Se desideri che tutti i responsabili IAM che accedono a una risorsa provengano da un membro Account AWS della tua organizzazione (incluso l'account di AWS Organizations gestione), puoi utilizzare la `aws:PrincipalOrgID` chiave global condition.

Per concedere o limitare questo tipo di accesso, definisci la condizione `aws:PrincipalOrgID` e imposta il valore sull'ID dell'organizzazione nella policy di bucket. L'ID dell'organizzazione viene utilizzato per controllare l'accesso al bucket. Quando si utilizza la condizione `aws:PrincipalOrgID`, le autorizzazioni della policy di bucket vengono applicate anche a tutti i nuovi account aggiunti all'organizzazione.

Ecco un esempio di policy di bucket basata su risorse che puoi utilizzare per concedere l'accesso diretto al bucket a specifici principali IAM nella tua organizzazione. Aggiungendo la chiave di condizione globale `aws:PrincipalOrgID` alla policy di bucket, ora l'account principale deve trovarsi nell'organizzazione per ottenere l'accesso alla risorsa. Anche se si specifica accidentalmente un account errato quando si concede l'accesso, la chiave di condizione globale aws:PrincipalOrgID funge da ulteriore protezione. Quando viene utilizzata come policy, questa chiave globale impedisce

a tutti i principali esterni all'organizzazione specificata di accedere al bucket S3. Solo i principali degli account dell'organizzazione elencata possono ottenere l'accesso alla risorsa.

```
{
    "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"]
            }
        }
    }]
}
```

Gestione dell'accesso in base a indirizzi IP specifici

Limitare l'accesso a indirizzi IP specifici

L'esempio seguente impedisce a tutti gli utenti di eseguire operazioni di Amazon S3 sugli oggetti nei bucket specificati, a meno che la richiesta non provenga dall'intervallo di indirizzi IP specificato.

> **Note**
>
> Quando limiti l'accesso a un indirizzo IP specifico, assicurati di specificare anche quali endpoint VPC, indirizzi IP di origine VPC o indirizzi IP esterni possono accedere al bucket S3. In caso contrario, potresti perdere l'accesso al bucket se la policy impedisce a tutti gli utenti sprovvisti delle autorizzazioni appropriate di eseguire operazioni S3 sugli oggetti del bucket.

L'istruzione `Condition` di questa policy identifica *`192.0.2.0/24`* come l'intervallo di indirizzi IP Internet Protocol versione 4 (IPv4) consentiti.

Il `Condition` blocco utilizza la `NotIpAddress` condizione e la chiave `aws:SourceIp` condition, che è una chiave di condizione AWS ampia. La chiave di condizione `aws:SourceIp` può essere

utilizzata solo per intervalli di indirizzi IP pubblici. Per ulteriori informazioni su queste chiavi di condizione, consulta Chiavi relative alle condizioni delle politiche per Amazon S3. I valori IPv4 `aws:SourceIp` utilizzano la notazione CIDR standard. Per ulteriori informazioni, consulta i riferimenti agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

> **Warning**
>
> Prima di utilizzare questa policy, sostituisci l'intervallo di indirizzi IP *192.0.2.0/24* riportato in questo esempio con un valore appropriato per il tuo caso d'uso. Altrimenti, perderai la possibilità di accedere al tuo bucket.

```
{
    "Version": "2012-10-17",
    "Id": "S3PolicyId1",
    "Statement": [
        {
            "Sid": "IPAllow",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ],
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": "192.0.2.0/24"
                }
            }
        }
    ]
}
```

Consentire entrambi gli indirizzi IPv4 e IPv6

Quando inizi a utilizzare gli indirizzi IPv6, è consigliabile aggiornare tutte le policy dell'organizzazione con gli intervalli di indirizzi IPv6 in aggiunta agli intervalli di indirizzi IPv4 esistenti. Ciò contribuirà a garantire che le policy continuino a funzionare durante la transizione a IPv6.

Nella policy del bucket di esempio seguente viene mostrato come combinare gli intervalli di indirizzi IPv4 e IPv6 per coprire tutti gli indirizzi IP validi dell'organizzazione. La policy di esempio permette l'accesso agli indirizzi IP di esempio *192.0.2.1* e *2001:DB8:1234:5678::1* e lo nega agli indirizzi *203.0.113.1* e *2001:DB8:1234:5678:ABCD::1*.

La chiave di condizione `aws:SourceIp` può essere utilizzata solo per intervalli di indirizzi IP pubblici. I valori IPv6 per `aws:SourceIp` devono essere nel formato CIDR standard. Per IPv6 è supportato l'utilizzo di `::` per rappresentare un intervallo di zeri (0), ad esempio `2001:DB8:1234:5678::/64`. Per ulteriori informazioni, consulta Operatori di condizione con indirizzo IP nella Guida per l'utente di IAM.

> **Warning**
>
> Sostituire gli intervalli di indirizzi IP in questo esempio con valori appropriati per il caso d'uso prima di utilizzare questa policy. In caso contrario, si potrebbe perdere la possibilità di accedere al bucket.

```
{
    "Id": "PolicyId2",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowIPmix",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ],
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "2001:DB8:1234:5678::/64"
                    ]
                },
                "NotIpAddress": {
                    "aws:SourceIp": [
                        "203.0.113.0/24",
```

```
                    "2001:DB8:1234:5678:ABCD::/80"
                ]
            }
        }
    }
  ]
}
```

Gestione dell'accesso in base a richieste HTTP o HTTPS

Limitare l'accesso solo alle richieste HTTPS

Se desideri impedire a potenziali aggressori di manipolare il traffico di rete, puoi utilizzare HTTPS (TLS) per consentire solo le connessioni crittografate limitando al contempo l'accesso al tuo bucket da parte delle richieste HTTP. Per determinare se la richiesta è HTTP o HTTPS, utilizza la chiave di condizione globale aws:SecureTransport nella policy di bucket S3. La chiave di condizione `aws:SecureTransport` controlla se una richiesta è stata inviata utilizzando HTTP.

Se una richiesta restituisce `true`, la richiesta è stata inviata tramite HTTP. Se la richiesta restituisce `false`, la richiesta è stata inviata tramite HTTP. Puoi quindi consentire o negare l'accesso al bucket in base allo schema di richiesta desiderato.

Nell'esempio seguente, la policy di bucket nega esplicitamente l'accesso alle richieste HTTP.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "RestrictToTLSRequestsOnly",
        "Action": "s3:*",
        "Effect": "Deny",
        "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        ],
        "Condition": {
            "Bool": {
                "aws:SecureTransport": "false"
            }
        },
        "Principal": "*"
    }]
}
```

Limitare l'accesso a un referer HTTP specifico

Supponi di avere un sito Web con il nome di dominio (*www.example.com* o *example.com*) con collegamenti a foto e video archiviati nel bucket denominato *example-s3-bucket*. Per impostazione predefinita, tutte le risorse Amazon S3 sono private, quindi solo chi le Account AWS ha create può accedervi.

Per consentire l'accesso in lettura a questi oggetti dal sito Web, è possibile aggiungere una policy di bucket che concede l'autorizzazione `s3:GetObject` con una condizione secondo cui la richiesta `GET` deve generare da pagine Web specifiche. La seguente policy limita le richieste utilizzando la condizione `StringLike` con la chiave di condizione `aws:Referer`.

```
{
  "Version":"2012-10-17",
  "Id":"HTTP referer policy example",
  "Statement":[
    {
      "Sid":"Allow only GET requests originating from www.example.com and
 example.com.",
      "Effect":"Allow",
      "Principal":"*",
      "Action":["s3:GetObject","s3:GetObjectVersion"],
      "Resource":"arn:aws:s3:::example-s3-bucket/*",
      "Condition":{
        "StringLike":{"aws:Referer":["http://www.example.com/*","http://example.com/
*"]}
      }
    }
  ]
}
```

Verifica che i browser utilizzati includano l'intestazione HTTP `referer` nella richiesta.

> **Warning**
>
> Ti consigliamo di procedere cautela quando utilizzi la chiave di condizione `aws:Referer`. È pericoloso includere un valore di intestazione di un referer pubblicamente noto. Parti non autorizzate possono utilizzare browser modificati o personalizzati per fornire qualsiasi valore `aws:Referer` scelto. Pertanto, non `aws:Referer` utilizzarlo per impedire a parti non autorizzate di effettuare richieste dirette AWS .

La chiave di condizione `aws:Referer` è disponibile solo per consentire ai clienti di proteggere i propri contenuti digitali, come i contenuti archiviati in Amazon S3, da riferimenti su siti di terze parti non autorizzate. Per ulteriori informazioni, consulta la sezione aws:Referer nella Guida per l'utente di IAM.

Gestione dell'accesso utente a cartelle specifiche

Concedere agli utenti l'accesso a cartelle specifiche

Supponiamo che tu stia cercando di concedere agli utenti l'accesso a una cartella specifica. Se l'utente IAM e il bucket S3 appartengono allo stesso gruppo Account AWS, puoi utilizzare una policy IAM per concedere all'utente l'accesso a una cartella di bucket specifica. Con questo approccio, non è necessario aggiornare la policy di bucket per concedere l'accesso. Puoi aggiungere la policy IAM a un ruolo IAM a cui possono passare più utenti.

Se l'identità IAM e il bucket S3 appartengono a parti diverse Account AWS, devi concedere l'accesso a più account sia nella policy IAM che nella policy del bucket. Per informazioni su come concedere l'accesso multi-account, consulta la sezione relativa al proprietario del bucket che concede autorizzazioni per il bucket multi-account.

La seguente policy di bucket di esempio concede a *JohnDoe* l'accesso completo a livello di console solo alla sua cartella (`home/`*JohnDoe*`/`). Creando una cartella `home` e concedendo le autorizzazioni appropriate ai tuoi utenti, puoi fare in modo che più utenti condividano un singolo bucket. Questa policy è composta da tre istruzioni `Allow`:

- *AllowRootAndHomeListingOfCompanyBucket*: consente all'utente (*JohnDoe*) di elencare gli oggetti al livello root del bucket *DOC-EXAMPLE-BUCKET* e nella cartella `home`. Questa istruzione consente inoltre all'utente di cercare in base al prefisso `home/` utilizzando la console.
- *AllowListingOfUserFolder*: consente all'utente (*JohnDoe*) di elencare tutti gli oggetti nella cartella `home/`*JohnDoe*`/` e nelle eventuali sottocartelle.
- *AllowAllS3ActionsInUserFolder*: consente all'utente di eseguire tutte le operazioni di Amazon S3 concedendo le autorizzazioni `Read`, `Write` e `Delete`. Le autorizzazioni sono limitate alla cartella principale del proprietario 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/*"]
        }
```

```
    ]
}
```

Gestione dell'accesso per i log degli accessi

Concedere l'accesso ad Application Load Balancer per abilitare i log degli accessi

Quando abiliti i log degli accessi per Application Load Balancer, devi specificare il nome del bucket S3 in cui il sistema di bilanciamento del carico archivierà i log. Il bucket deve avere una policy collegata che concede a Elastic Load Balancing l'autorizzazione a scrivere nel bucket.

Nell'esempio seguente, la policy di bucket concede a Elastic Load Balancing (ELB) l'autorizzazione a scrivere i log degli accessi nel bucket:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": "arn:aws:iam::elb-account-id:root"
      },
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::example-s3-bucket/prefix/AWSLogs/111122223333/*"
    }
  ]
}
```

> **Note**
>
> Assicurati di sostituire *elb-account-id* con l'ID Account AWS per Elastic Load Balancing per la tua Regione AWS. Per l'elenco delle regioni Elastic Load Balancing, consulta Collegamento di una policy al bucket Amazon S3 nella Guida per l'utente di Elastic Load Balancing.

Se la tua Regione AWS non compare nell'elenco delle regioni Elastic Load Balancing supportate, utilizza la seguente politica, che concede le autorizzazioni al servizio di consegna dei log specificato.

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
       "Principal": {
         "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
          },
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::example-s3-bucket/prefix/AWSLogs/111122223333/*"
    }
  ]
}
```

Quindi, assicurati di configurare i log degli accessi di Elastic Load Balancing abilitandoli. Puoi verificare le autorizzazioni del bucket creando un file di test.

Gestione dell'accesso a un Amazon CloudFront OAI

Concedi l'autorizzazione a un Amazon CloudFront OAI

L'esempio seguente di bucket policy concede un'autorizzazione OAI (CloudFront Origin Access Identity) per ottenere (leggere) tutti gli oggetti nel bucket S3. Puoi utilizzare un CloudFront OAI per consentire agli utenti di accedere agli oggetti nel tuo bucket tramite Amazon S3 CloudFront , ma non direttamente. Per ulteriori informazioni, consulta Limitazione dell'accesso ai contenuti di Amazon S3 utilizzando un'identità di accesso di origine nella CloudFront Amazon Developer Guide.

La policy seguente utilizza l'ID dell'identità di accesso origine (OAI) come `Principal` della policy. Per ulteriori informazioni sull'utilizzo delle policy dei bucket S3 per concedere l'accesso a un CloudFront OAI, consulta Migrating from Origin Access Identity (OAI) a Origin Access Control (OAC) nella Amazon Developer Guide. CloudFront

Per utilizzare questo esempio:

- Sostituisci *`EH1HDMB1FH2TC`* con l'ID dell'identità di accesso origine (OAI). Per trovare l'ID dell'OAI, consulta la pagina Origin Access Identity sulla CloudFront console o utilizzalo nell'API. ListCloudFrontOriginAccessIdentities CloudFront
- Sostituisci *`example-s3-bucket`* con il nome del tuo 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:::example-s3-bucket/*"
        }
    ]
}
```

Gestione dell'accesso per Amazon S3 Storage Lens

Concedere le autorizzazioni per Amazon S3 Storage Lens

S3 Storage Lens aggrega i tuoi parametri e mostra le informazioni nella sezione Account snapshot (Snapshot dell'account) nella pagina Buckets (Bucket) della console di Amazon S3. S3 Storage Lens dispone inoltre di un pannello di controllo interattivo che puoi usare per visualizzare informazioni dettagliate e tendenze, contrassegnare le anomalie e ricevere consigli per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati. Nel pannello di controllo sono disponibili opzioni di drill-down per generare e visualizzare approfondimenti a livello di organizzazione, account, Regione AWS, classe di archiviazione, bucket, prefisso o gruppo Storage Lens. Puoi anche inviare un'esportazione dei parametri quotidiana in formato CSV o Parquet a un bucket S3.

S3 Storage Lens può esportare i parametri aggregati relativi l'utilizzo dell'archiviazione in un bucket Amazon S3 per ulteriori analisi. Il bucket in cui S3 Storage Lens colloca le esportazioni delle metriche è noto come bucket di destinazione. Quando configuri l'esportazione delle metriche di S3 Storage Lens, devi disporre di una policy di bucket per il bucket di destinazione. Per ulteriori informazioni, consulta Valutazione dell'attività e dell'utilizzo dello storage con Amazon S3 Storage Lens.

La seguente policy di bucket di esempio concede ad Amazon S3 l'autorizzazione a scrivere oggetti (richieste PUT) in un bucket di destinazione. Questo tipo di policy di bucket viene utilizzato nel bucket di destinazione quando si imposta l'esportazione dei parametri di S3 Storage Lens.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3StorageLensExamplePolicy",
```

```
            "Effect": "Allow",
            "Principal": {
                "Service": "storage-lens.s3.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::destination-bucket/destination-prefix/
StorageLens/111122223333/*"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": "111122223333",
                    "aws:SourceArn": "arn:aws:s3:region-code:111122223333:storage-
lens/storage-lens-dashboard-configuration-id"
                }
            }
        }
    ]
}
```

Utilizza la modifica seguente alla precedente istruzione `Resource` della policy di bucket quando configuri un'esportazione di parametri a livello di organizzazione S3 Storage Lens.

```
"Resource": "arn:aws:s3:::destination-bucket/destination-prefix/StorageLens/your-
organization-id/*",
```

Gestione delle autorizzazioni per i report di S3 Inventory, S3 Analytics e S3 Inventory

Concedere autorizzazioni per S3 Inventory e S3 Analytics

S3 Inventory crea elenchi di oggetti in un bucket, mentre l'esportazione di analisi della classe di archiviazione di S3 Analytics genera file di output dei dati utilizzati nell'analisi. Il bucket per il quale l'inventario elenca gli oggetti è denominato bucket di origine. Il bucket nel quale viene scritto il file di inventario e il file di esportazione di analisi è definito bucket di destinazione. È necessario creare una policy di bucket per il bucket di destinazione quando si configura un inventario o un'esportazione di analisi. Per ulteriori informazioni, consulta Amazon S3 Inventory e Analisi di Amazon S3 – Analisi della classe di storage.

La policy di bucket di esempio seguente concede ad Amazon S3 l'autorizzazione per scrivere oggetti (richieste PUT) dall'account per il bucket di origine nel bucket di destinazione. Questo tipo di policy di bucket viene utilizzato nel bucket di destinazione quando imposti S3 Inventory e l'esportazione di S3 Analytics.

```
{
      "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"
                }
            }
        }
    ]
}
```

Controlla la creazione della configurazione dei report di S3 Inventory

Amazon S3 Inventory crea elenchi degli oggetti presenti in un bucket S3 e i metadata per ogni oggetto. L'`s3:PutInventoryConfiguration`autorizzazione consente a un utente di creare una configurazione di inventario che includa tutti i campi di metadati degli oggetti disponibili per impostazione predefinita e di specificare il bucket di destinazione in cui archiviare l'inventario. Un utente con accesso in lettura agli oggetti nel bucket di destinazione può accedere a tutti i campi di metadati degli oggetti disponibili nel report di inventario. Per ulteriori informazioni sui campi dei metadati disponibili in S3 Inventory, consulta Elenco di Amazon S3 Inventory.

Per impedire a un utente di configurare un rapporto S3 Inventory, rimuovi l'`s3:PutInventoryConfiguration`autorizzazione all'utente.

Alcuni campi di metadati degli oggetti nelle configurazioni dei report di S3 Inventory sono facoltativi, il che significa che sono disponibili per impostazione predefinita, ma possono essere limitati quando concedi l'autorizzazione a un utente. `s3:PutInventoryConfiguration` Puoi controllare se gli utenti possono includere questi campi di metadati opzionali nei loro report utilizzando la chiave di condizione. `s3:InventoryAccessibleOptionalFields` Per un elenco dei campi di metadati opzionali disponibili in S3 Inventory, consulta OptionalFieldsAmazon Simple Storage Service API Reference.

Per concedere a un utente l'autorizzazione a creare una configurazione di inventario con campi di metadati opzionali specifici, utilizza la chiave `s3:InventoryAccessibleOptionalFields` condition per rifinire le condizioni della tua bucket policy.

La seguente politica di esempio concede a un utente (*Ana*) l'autorizzazione a creare una configurazione di inventario in modo condizionale. La `ForAllValues:StringEquals` condizione nella politica utilizza la chiave `s3:InventoryAccessibleOptionalFields` condition per specificare i due campi di metadati opzionali consentiti, vale a dire e. `Size` `StorageClass` Pertanto, quando crea *Ana* una configurazione di inventario, gli unici campi di metadati opzionali che può includere sono `Size` e. `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"
        ]
```

```
      }
    }
   }
 ]
}
```

Per impedire a un utente di configurare un report S3 Inventory che includa campi di metadati opzionali specifici, aggiungi una `Deny` dichiarazione esplicita alla policy del bucket di origine. L'esempio seguente di bucket policy impedisce all'utente di creare una configurazione di inventario nel *DOC-EXAMPLE-SOURCE-BUCKET* bucket di origine che *Ana* includa i campi opzionali o di metadati. `ObjectAccessControlList` `ObjectOwner` L'utente *Ana* può comunque creare una configurazione di inventario con altri campi di metadati opzionali.

```
{
 "Id": "InventoryConfigSomeFields",
 "Version": "2012-10-17",
 "Statement": [{
   "Sid": "AllowInventoryCreation",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/Ana"
   },
   "Action": "s3:PutInventoryConfiguration",
   "Resource":
    "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET",

  },
  {
   "Sid": "DenyCertainInventoryFieldCreation",
   "Effect": "Deny",
   "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/Ana"
   },
   "Action": "s3:PutInventoryConfiguration",
   "Resource":
     "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET",
   "Condition": {
    "ForAnyValue:StringEquals": {
     "s3:InventoryAccessibleOptionalFields": [
        "ObjectOwner",
        "ObjectAccessControlList"
        ]
```

```
      }
    }
   }
 ]
}
```

> **Note**
>
> L'uso della chiave di `s3:InventoryAccessibleOptionalFields` condizione nelle politiche relative ai bucket non influisce sulla fornitura di report di inventario basati sulle configurazioni di inventario esistenti.

> **Important**
>
> Si consiglia di utilizzarlo `ForAllValues` con un `Allow` effetto o `ForAnyValue` con un `Deny` effetto, come mostrato negli esempi precedenti.
> Non utilizzare `ForAllValues` con un `Deny` effetto né `ForAnyValue` con un `Allow` effetto, poiché queste combinazioni possono essere eccessivamente restrittive e bloccare l'eliminazione della configurazione dell'inventario.
> Per saperne di più sugli operatori *`ForAllValues`* e sui set di *`ForAnyValue`* condizioni, consulta le chiavi di contesto multivalore nella Guida per l'utente IAM.

Richiesta dell'autenticazione a più fattori (MFA)

Amazon S3 supporta l'accesso all'API protetto con autenticazione MFA, una caratteristica che permette di imporre la Multi-Factor Authentication (MFA) per accedere alle risorse di Amazon S3. L'autenticazione a più fattori offre un ulteriore livello di sicurezza che puoi applicare al tuo ambiente. AWS L'autenticazione a più fattori (MFA) è una funzione di protezione che prevede che gli utenti dimostrino di possedere fisicamente un dispositivo MFA fornendo un codice MFA valido. Per ulteriori informazioni, consulta Autenticazione a più fattori (MFA) di AWS. Puoi richiedere l'autenticazione MFA per tutte le richieste di accesso alle risorse di Amazon S3.

Per imporre l'uso del requisito dell'autenticazione a più fattori (MFA), utilizza la chiave di condizione `aws:MultiFactorAuthAge` in una policy di bucket. Gli utenti IAM possono accedere alle risorse Amazon S3 utilizzando credenziali temporanee emesse da (). AWS Security Token Service AWS STS Al momento della richiesta AWS STS , dovrai fornire il codice MFA.

Quando Amazon S3 riceve una richiesta con l'autenticazione a più fattori (MFA), la chiave di condizione `aws:MultiFactorAuthAge` fornisce un valore numerico che indica il tempo trascorso (in secondi) dalla creazione delle credenziali temporanee. Se le credenziali temporanee fornite nella richiesta non sono state create utilizzando un dispositivo MFA, il valore di questa chiave è null (assente). In una policy di bucket, è possibile aggiungere una condizione per controllare questo valore, come mostrato nell'esempio riportato di seguito.

La policy di esempio nega qualsiasi operazione Amazon S3 nella cartella */taxdocuments* del bucket *example-s3-bucket* se la richiesta non è autenticata tramite l'autenticazione a più fattori (MFA). Per ulteriori informazioni su MFA, consulta la sezione Utilizzo dell'autenticazione a più fattori (MFA) in AWS nella Guida per l'utente di IAM.

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::example-s3-bucket/taxdocuments/*",
        "Condition": { "Null": { "aws:MultiFactorAuthAge": true }}
      }
    ]
 }
```

La condizione `Null` nel blocco `Condition` viene valutata come `true` se il valore della chiave di condizione `aws:MultiFactorAuthAge` è null, a indicare che le credenziali di sicurezza temporanee nella richiesta sono state create senza un dispositivo MFA.

La policy di bucket seguente è un'estensione di quella precedente. La seguente policy sui bucket include due dichiarazioni politiche. Una dichiarazione consente l'autorizzazione `s3:GetObject` per un bucket (*example-s3-bucket*) per tutti gli utenti. La seconda dichiarazione limita ulteriormente l'accesso alla cartella *example-s3-bucket/taxdocuments* nel bucket richiedendo l'autenticazione MFA.

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
```

```
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::example-s3-bucket/taxdocuments/*",
        "Condition": { "Null": { "aws:MultiFactorAuthAge": true } }
      },
      {
        "Sid": "",
        "Effect": "Allow",
        "Principal": "*",
        "Action": ["s3:GetObject"],
        "Resource": "arn:aws:s3:::example-s3-bucket/*"
      }
    ]
 }
```

Facoltativamente, è possibile utilizzare una condizione numerica per limitare la durata della validità della chiave `aws:MultiFactorAuthAge`. La durata specificata con la chiave `aws:MultiFactorAuthAge` è indipendente dalla durata delle credenziali di sicurezza temporanee utilizzate per l'autenticazione della richiesta.

Ad esempio, la policy di bucket seguente, oltre a richiedere l'autenticazione MFA, controlla anche da quanto tempo esiste la sessione temporanea. La policy rifiuta tutte le operazioni se il valore della chiave `aws:MultiFactorAuthAge` indica che la sessione temporanea è stata creata oltre un'ora (3.600 secondi) prima.

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::example-s3-bucket/taxdocuments/*",
        "Condition": {"Null": {"aws:MultiFactorAuthAge": true }}
      },
      {
        "Sid": "",
```

```
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::example-s3-bucket/taxdocuments/*",
        "Condition": {"NumericGreaterThan": {"aws:MultiFactorAuthAge": 3600 }}
      },
      {
        "Sid": "",
        "Effect": "Allow",
        "Principal": "*",
        "Action": ["s3:GetObject"],
        "Resource": "arn:aws:s3:::example-s3-bucket/*"
      }
    ]
}
```

Impedire agli utenti di eliminare oggetti

Per default, gli utenti non dispongono di autorizzazioni. Tuttavia, mentre crei le politiche, potresti concedere agli utenti autorizzazioni che non intendevi concedere. Per evitare tali lacune nelle autorizzazioni, puoi scrivere una politica di accesso più rigorosa aggiungendo un rifiuto esplicito.

Per impedire in modo esplicito agli utenti o agli account di eliminare oggetti, è necessario aggiungere le seguenti azioni a una policy bucket: e le autorizzazioni. `s3:DeleteObject` `s3:DeleteObjectVersion` `s3:PutLifecycleConfiguration` Tutte e tre le azioni sono necessarie perché puoi eliminare gli oggetti chiamando esplicitamente l'API DELETE Object o configurando il loro ciclo di vita (vediGestione del ciclo di vita dello storage) in modo che Amazon S3 possa rimuovere gli oggetti alla scadenza del loro ciclo di vita.

Nel seguente esempio di policy, neghi esplicitamente le autorizzazioni DELETE Object all'utente Dave. Una negazione esplicita sostituisce sempre qualsiasi altra autorizzazione concessa.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Dave"
      },
      "Action": [
```

```
        "s3:GetObjectVersion",
        "s3:GetBucketAcl"
      ],
      "Resource": [
        "arn:aws:s3:::example-s3-bucket1",
   "arn:aws:s3:::example-s3-bucket1/*"
      ]
    },
    {
      "Sid": "statement2",
      "Effect": "Deny",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Dave"
      },
      "Action": [
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:PutLifecycleConfiguration"
      ],
      "Resource": [
        "arn:aws:s3:::example-s3-bucket1",
     "arn:aws:s3:::example-s3-bucket1/*"
      ]
    }
  ]
}
```

Esempi di policy Bucket che utilizzano chiavi condizionali

È possibile utilizzare in linguaggio delle policy di accesso per specificare le condizioni quando si concedono le autorizzazioni. È possibile utilizzare l'elemento `Condition` facoltativo o il blocco `Condition` per specificare le condizioni per l'applicazione di una policy.

Per le policy che utilizzano le chiavi di condizioni di Amazon S3 per operazioni su oggetti e bucket, consulta gli esempi seguenti. Per ulteriori informazioni su queste chiavi di condizione, consulta Chiavi relative alle condizioni delle politiche per Amazon S3. Per un elenco completo di azioni, chiavi di condizione e risorse di Amazon S3 che puoi specificare nelle politiche, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.

Esempi – Chiavi di condizione di Amazon S3 per le operazioni sugli oggetti

In questa sezione vengono forniti esempi che illustrano come utilizzare le chiavi di condizione specifiche di Amazon S3 per le operazioni sugli oggetti. Per un elenco completo di azioni, chiavi di

condizione e risorse di Amazon S3 che puoi specificare nelle politiche, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.

Molte delle policy di esempio mostrano come è possibile utilizzare le chiavi di condizione con le operazioni PUT Object. Le operazioni PUT Object permettono intestazioni specifiche della lista di controllo degli accessi (ACL) che è possibile utilizzare per concedere autorizzazioni basate sulle liste ACL. Utilizzando queste chiavi, il proprietario del bucket può impostare una condizione per richiedere determinate autorizzazioni di accesso specifiche quando l'utente carica un oggetto. Puoi anche concedere autorizzazioni basate su ACL con l'operazione. PutObjectAcl Per ulteriori informazioni, consulta il riferimento PutObjectAclall'API Amazon S3 Amazon Simple Storage Service. Per ulteriori informazioni sulle ACL, consulta Panoramica delle liste di controllo accessi (ACL).

Argomenti

- Esempio 1: Concessione di s3: PutObject autorizzazione che richiede l'autorizzazione agli oggetti archiviati utilizzando la crittografia lato server
- Esempio 2: Concessione a s3: PutObject autorizzazione a copiare oggetti con una restrizione sulla fonte di copia
- Esempio 3: Concessione dell'accesso a una versione specifica di un oggetto
- Esempio 4: concessione di autorizzazioni basate sui tag degli oggetti
- Esempio 5: limitazione dell'accesso in base all' Account AWS ID del proprietario del bucket
- Esempio 6: Richiesta di una versione TLS minima

Esempio 1: Concessione di s3: PutObject autorizzazione che richiede l'autorizzazione agli oggetti archiviati utilizzando la crittografia lato server

Si supponga che l'Account A possieda un bucket. L'amministratore dell'account vuole concedere a Jane, un'utente dell'Account A, l'autorizzazione per il caricamento di oggetti con la condizione che Jane richieda sempre la crittografia lato server in modo che Amazon S3 salvi gli oggetti crittografati. L'amministratore dell'Account A può procedere utilizzando la chiave di condizione `s3:x-amz-server-side-encryption` come mostrato. La coppia chiave-valore nel blocco `Condition` specifica la chiave `s3:x-amz-server-side-encryption`.

```
"Condition": {
     "StringNotEquals": {
         "s3:x-amz-server-side-encryption": "AES256"
     }}
```

Quando si verifica l'autorizzazione utilizzando il AWS CLI, è necessario aggiungere il parametro richiesto utilizzando il parametro. `--server-side-encryption`

```
aws s3api put-object --bucket example1bucket --key HappyFace.jpg --body c:
\HappyFace.jpg --server-side-encryption "AES256" --profile AccountBadmin
```

Esempio 2: Concessione a s3: PutObject autorizzazione a copiare oggetti con una restrizione sulla fonte di copia

Nella richiesta PUT Object, quando specifichi un oggetto di origine, viene eseguita un'operazione di copia (consulta PUT Object - Copy). Di conseguenza è possibile che il proprietario del bucket assegni all'utente l'autorizzazione a copiare gli oggetti con qualche limitazione sull'origine, ad esempio:

- Consentire la copia di oggetti solo dal bucket `sourcebucket`.
- Consentire la copia di oggetti dal bucket di origine e solo di quegli oggetti il cui prefisso del nome della chiave inizia con public/ f (ad esempio, sourcebucket/public/*).
- Consentire la copia di un solo oggetto specifico dal bucket di origine (ad esempio, sourcebucket/example.jpg).

La policy del bucket seguente concede all'utente (Dave) l'autorizzazione `s3:PutObject` che gli consente di copiare solo gli oggetti con la condizione che la richiesta includa l'intestazione `s3:x-amz-copy-source` e il valore di intestazione specifichi il prefisso del nome della chiave `/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/*"
                }
            }
        }
    ]
}
```

Prova la politica con AWS CLI

È possibile verificare l'autorizzazione utilizzando il AWS CLI `copy-object` comando. È possibile specificare l'origine aggiungendo il parametro `--copy-source`; il prefisso del nome della chiave deve corrispondere al prefisso consentito nella policy. È necessario fornire le credenziali all'utente Dave utilizzando il parametro `--profile`. Per ulteriori informazioni sulla configurazione di AWS CLI, vedereSviluppo con Amazon S3 tramite la AWS CLI.

```
aws s3api copy-object --bucket awsexamplebucket1 --key HappyFace.jpg
--copy-source examplebucket/public/PublicHappyFace1.jpg --profile AccountADave
```

Concessione dell'autorizzazione a copiare solo un oggetto specifico

La policy di cui sopra utilizza la condizione `StringNotLike`. Per assegnare l'autorizzazione a copiare solo un determinato oggetto, è necessario modificare la condizione da `StringNotLike` a `StringNotEquals` e quindi specificare l'esatta chiave dell'oggetto, come illustrato.

```
"Condition": {
       "StringNotEquals": {
           "s3:x-amz-copy-source": "awsexamplebucket1/public/PublicHappyFace1.jpg"
       }
}
```

Esempio 3: Concessione dell'accesso a una versione specifica di un oggetto

Si supponga che l'Account A possieda un bucket abilitato per le versioni. Il bucket ha diverse versioni dell'oggetto `HappyFace.jpg`. L'amministratore dell'account ora vuole assegnare al suo utente

Dave l'autorizzazione a ottenere unicamente una versione specifica dell'oggetto. L'amministratore dell'account può procedere assegnando a Dave l'autorizzazione `s3:GetObjectVersion` in base a condizioni, come mostrato di seguito. La coppia chiave-valore nel blocco `Condition` specifica la chiave di condizione `s3:VersionId`. In questo caso, Dave deve conoscere l'esatto ID di versione dell'oggetto per poterlo recuperare.

Per ulteriori informazioni, consulta GetObjectAmazon Simple Storage Service API Reference.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": "s3:GetObjectVersion",
            "Resource": "arn:aws:s3:::examplebucketversionenabled/HappyFace.jpg"
        },
        {
            "Sid": "statement2",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": "s3:GetObjectVersion",
            "Resource": "arn:aws:s3:::examplebucketversionenabled/HappyFace.jpg",
            "Condition": {
                "StringNotEquals": {
                    "s3:VersionId": "AaaHbAQitwiL_h47_44lRO2DDfLlBO5e"
                }
            }
        }
    ]
}
```

Testa la policy con il AWS CLI

È possibile testare le autorizzazioni utilizzando il AWS CLI `get-object` comando con il `--version-id` parametro che identifica la versione specifica dell'oggetto. Il comando recupera l'oggetto e lo salva nel file `OutputFile.jpg`.

```
aws s3api get-object --bucket examplebucketversionenabled --key HappyFace.jpg
 OutputFile.jpg --version-id AaaHbAQitwiL_h47_44lRO2DDfLlBO5e --profile AccountADave
```

Esempio 4: concessione di autorizzazioni basate sui tag degli oggetti

Per esempi su come utilizzare le chiavi di condizione per il tagging di oggetti con le operazioni di Amazon S3, consulta Tagging e policy di controllo degli accessi.

Esempio 5: limitazione dell'accesso in base all' Account AWS ID del proprietario del bucket

Puoi utilizzare la chiave di condizione `aws:ResourceAccount` o `s3:ResourceAccount` per scrivere policy IAM o di endpoint del cloud privato virtuale (VPC) che limitano l'accesso di utenti, ruoli o applicazioni ai bucket Amazon S3 di proprietà di un determinato ID dell' Account AWS . Ciò è utile se desideri impedire ai client all'interno del tuo VPC di accedere a bucket di cui non sei proprietario.

Tuttavia, tieni presente che alcuni AWS servizi si basano sull'accesso a bucket AWS gestiti. Pertanto, l'utilizzo della chiave di condizione `aws:ResourceAccount` o `s3:ResourceAccount` nella policy IAM potrebbe anche influire sull'accesso a tali risorse.

Per ulteriori informazioni ed esempi, consulta le seguenti risorse:

- Limitazione dell'accesso ai bucket in un Account AWS specificato nella Guida di AWS PrivateLink
- Limitazione dell'accesso ai bucket utilizzati da Amazon ECRnella Guida di Amazon ECR
- Fornisci l'accesso richiesto a Systems Manager per i bucket Amazon S3 AWS gestiti nella guida AWS Systems Manager
- Limitazione dell'accesso ai bucket Amazon S3 di proprietà di specifici Account AWS nel Blog di archiviazione AWS

Esempio 6: Richiesta di una versione TLS minima

Puoi utilizzare s3: TlsVersion condition key per scrivere policy IAM, Virtual Private Cloud Endpoint (VPCE) o bucket che limitano l'accesso di utenti o applicazioni ai bucket Amazon S3 in base alla versione TLS utilizzata dal client. È possibile utilizzare questa chiave di condizione per scrivere policy che richiedono una versione TLS minima.

Example

Questo esempio di bucket policy nega le PutObject richieste dei client con una versione TLS inferiore alla 1.2, ad esempio 1.1 o 1.0.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::example-s3-bucket1",
                "arn:aws:s3:::example-s3-bucket1/*"
            ],
            "Condition": {
                "NumericLessThan": {
                    "s3:TlsVersion": 1.2
                }
            }
        }
    ]
}
```

Example

Questo esempio di bucket policy consente PutObject le richieste da parte di client con una versione TLS superiore alla 1.1, ad esempio 1.2, 1.3 o successiva.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::example-s3-bucket1",
                "arn:aws:s3:::example-s3-bucket1/*"
            ],
            "Condition": {
                "NumericGreaterThan": {
                    "s3:TlsVersion": 1.1
                }
```

```
            }
        }
    ]
}
```

Esempi – Chiavi di condizione di Amazon S3 per le operazioni sui bucket

In questa sezione vengono fornite policy di esempio che illustrano come utilizzare le chiavi di condizione specifiche di Amazon S3 per le operazioni sui bucket.

Argomenti

- Esempio 1: Concessione di s3: GetObject autorizzazione con una condizione su un indirizzo IP
- Esempio 2: recupero di un elenco di oggetti in un bucket con un prefisso specifico
- Esempio 3: impostazione del numero massimo di chiavi

Esempio 1: Concessione di s3: GetObject autorizzazione con una condizione su un indirizzo IP

È possibile concedere agli utenti autenticati il permesso di utilizzare l'`s3:GetObject`azione se la richiesta proviene da un intervallo specifico di indirizzi IP (192.0.2.*), a meno che l'indirizzo IP non sia 192.0.2.188. Nel blocco di condizioni, `IpAddress` e `NotIpAddress` sono condizioni e per ogni condizione viene fornita una coppia chiave-valore per la valutazione. Entrambe le coppie chiave-valore in questo esempio utilizzano la chiave -wide. `aws:SourceIp` AWS

> **Note**
>
> I valori chiave `IPAddress` e `NotIpAddress` specificati nella condizione utilizzano la notazione CIDR come descritto in RFC 4632. Per ulteriori informazioni, consulta 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"
                }
            }
        }
    ]
}
```

Puoi anche utilizzare altre chiavi di condizione AWS a livello di -wide nelle policy di Amazon S3. Ad esempio, è possibile specificare le chiavi di condizione `aws:SourceVpce` e `aws:SourceVpc` nelle policy di bucket per gli endpoint VPC. Per esempi specifici consulta Controllo dell'accesso dagli endpoint VPC con policy di bucket.

> **Note**
>
> Per alcune chiavi di condizione AWS globali, sono supportati solo determinati tipi di risorse. Pertanto, verificare se Amazon S3 supporta la chiave di condizione globale e il tipo di risorsa che si desidera utilizzare o se sarà invece necessario utilizzare una chiave di condizione specifica per Amazon S3. Per un elenco completo dei tipi di risorse e delle chiavi di condizione supportati per Amazon S3, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.

Esempio 2: recupero di un elenco di oggetti in un bucket con un prefisso specifico

Puoi utilizzare la chiave `s3:prefix` condition per limitare la risposta dell'API GET Bucket (ListObjects) ai nomi di chiave con un prefisso specifico. Se si è il proprietario del bucket, è possibile limitare un utente a elencare il contenuto di un prefisso specifico nel bucket. Questa chiave di condizione risulta utile se gli oggetti nel bucket sono organizzati per prefissi del nomi delle chiavi. La console di Amazon S3 utilizza i prefissi dei nomi delle chiavi per mostrare un concetto di cartella. Solo la console supporta il concetto di cartelle, mentre l'API Amazon S3 supporta solo bucket e oggetti. Per ulteriori informazioni sull'utilizzo di prefissi e delimitatori per filtrare le autorizzazioni di accesso, consulta Procedura guidata: controllo dell'accesso a un bucket con policy utente.

Ad esempio, se vi sono due oggetti con nomi delle chiavi `public/object1.jpg` e `public/object2.jpg`, la console mostra gli oggetti nella cartella `public`. Nell'API Amazon S3 questi sono oggetti con prefissi, non oggetti nelle cartelle. Se tuttavia nell'API Amazon S3 organizzi le chiavi degli oggetti utilizzando tali prefissi, puoi concedere l'autorizzazione `s3:ListBucket` con la condizione `s3:prefix` che permette all'utente di ottenere un elenco dei nomi delle chiavi con un tali prefissi specifici.

In questo esempio, il proprietario del bucket e l'account padre a cui appartiene l'utente corrispondono. Quindi, il proprietario del bucket può utilizzare una policy di bucket o una policy utente. Per ulteriori informazioni su altre chiavi condizionali che puoi utilizzare con l'API GET Bucket (ListObjects), consulta. ListObjects

Policy utente

La policy utente seguente concede l'autorizzazione `s3:ListBucket` (consulta GET Bucket (List Objects)) con una condizione che richiede all'utente di specificare `prefix` nella richiesta con il valore `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 condizione limita l'utente a elencare solo le chiavi degli oggetti con il prefisso `projects`. Il rifiuto esplicito aggiunto rifiuta all'utente la richiesta di elencazione delle chiavi con qualsiasi altro prefisso, indipendentemente da quale altra autorizzazione possa avere l'utente. Ad esempio, è possibile che l'utente ottenga l'autorizzazione a elencare le chiavi degli oggetti senza alcuna limitazione in base agli aggiornamenti alla precedente policy utente oppure tramite una policy di bucket. Poiché il rifiuto esplicito è sempre prevalente, la richiesta dell'utente di elencare chiavi diverse dal prefisso `projects` viene rifiutata.

Policy del bucket

Se si aggiunge l'elemento `Principal` alla policy utente precedente, identificando l'utente, si ottiene una policy di bucket come illustrato.

```
{
   "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"
                }
             }
        }
    ]
}
```

Testate la politica con AWS CLI

È possibile testare la politica utilizzando il seguente `list-object` AWS CLI comando. Nel comando, vengono fornite le credenziali utente utilizzando il parametro `--profile`. Per ulteriori informazioni sulla configurazione e l'utilizzo di AWS CLI, vedereSviluppo con Amazon S3 tramite la AWS CLI.

```
aws s3api list-objects --bucket awsexamplebucket1 --prefix examplefolder --profile
 AccountADave
```

Se il bucket è abilitato per le versioni, al fine di elencare gli oggetti nel bucket invece dell'autorizzazione `s3:ListBucket` è necessario assegnare l'autorizzazione `s3:ListBucketVersions` nella policy precedente. Questa autorizzazione supporta la chiave di condizione `s3:prefix`.

Esempio 3: impostazione del numero massimo di chiavi

È possibile utilizzare la chiave `s3:max-keys` condition per impostare il numero massimo di chiavi che il richiedente può restituire in un GET Bucket (ListObjects) o ListObjectVersionsin una richiesta. Per impostazione predefinita, l'API restituisce fino a 1.000 chiavi. Per un elenco di operatori di condizione numerici che è possibile utilizzare con `s3:max-keys` e i relativi esempi, consulta Operatori di condizione numerici nella Guida per l'utente di IAM.

Policy basate sull'identità per Amazon S3

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse Amazon S3. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS API. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Per dettagli sulle azioni e sui tipi di risorse definiti da Amazon S3, incluso il formato degli ARN per ciascun tipo di risorsa, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.

Argomenti

- Best practice per le policy
- Esempi di policy basate sull'identità per Amazon S3
- Procedura guidata: controllo dell'accesso a un bucket con policy utente

Best practice per le policy

Le policy basate sull'identità determinano se qualcuno può creare, accedere o eliminare risorse Amazon S3 nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni.AWS Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta Policy gestite da AWSo Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse puoi aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente IAM.

- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per IAM Access Analyzer nella Guida per l'utente IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Configurazione dell'accesso alle API protetto con MFA nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Esempi di policy basate sull'identità per Amazon S3

Questa sezione mostra diversi esempi di policy basate sull'identità AWS Identity and Access Management (IAM) per il controllo dell'accesso ad Amazon S3. Ad esempio, le bucket policy (politiche basate sulle risorse), vedi. Politiche Bucket per Amazon S3 Per informazioni sul linguaggio delle policy IAM, consulta Politiche e autorizzazioni in Amazon S3.

Le policy di esempio seguenti funzionano se vengono testate a livello di programma. Per utilizzarle con la console di Amazon S3 occorre tuttavia concedere autorizzazioni aggiuntive che sono richieste dalla console. Per ulteriori informazioni sull'utilizzo di policy come queste con la console di Amazon S3, consulta Procedura guidata: controllo dell'accesso a un bucket con policy utente.

Argomenti

- Concessione a un utente IAM dell'accesso a uno dei bucket
- Concessione a ogni utente IAM dell'accesso a una cartella in un bucket
- Concessione a un gruppo dell'accesso a una cartella condivisa in Amazon S3
- Permesso a tutti gli utenti di leggere gli oggetti in una parte del bucket
- Permesso a un partner di rilasciare i file in una parte specifica del bucket
- Restrizione dell'accesso ai bucket Amazon S3 in un Account AWS specifico
- Limitazione dell'accesso ai bucket Amazon S3 all'interno dell'unità organizzativa
- Limitazione dell'accesso ai bucket Amazon S3 all'interno dell'organizzazione

- Concessione del permesso di recuperare la PublicAccessBlock configurazione per un Account AWS
- Limitazione della creazione di bucket a una sola regione

Concessione a un utente IAM dell'accesso a uno dei bucket

In questo esempio, vuoi concedere a un utente IAM incluso nel tuo account l' Account AWS accesso a uno dei tuoi bucket, *example-s3-bucket1*, e consentire all'utente di aggiungere, aggiornare ed eliminare oggetti.

Oltre ad assegnare le autorizzazioni `s3:PutObject`, `s3:GetObject` e `s3:DeleteObject` all'utente, la policy assegna anche le autorizzazioni `s3:ListAllMyBuckets`, `s3:GetBucketLocation` e `s3:ListBucket`. Queste sono le autorizzazioni aggiuntive richieste dalla console. Inoltre, le operazioni `s3:PutObjectAcl` e `s3:GetObjectAcl` sono necessarie per essere in grado di copiare, tagliare e incollare gli oggetti nella console. Per una procedura guidata di esempio che concede autorizzazioni a utenti e le testa utilizzando la console, consulta Procedura guidata: controllo dell'accesso a un bucket con policy utente.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action": "s3:ListAllMyBuckets",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":["s3:ListBucket","s3:GetBucketLocation"],
         "Resource":"arn:aws:s3:::example-s3-bucket1"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:GetObject",
            "s3:GetObjectAcl",
            "s3:DeleteObject"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket1/*"
```

```
      }
   ]
}
```

Concessione a ogni utente IAM dell'accesso a una cartella in un bucket

In questo esempio, vuoi che due utenti IAM, Mary e Carlos, abbiano accesso al tuo bucket, *example-s3-bucket1, in modo che possano aggiungere*, aggiornare ed eliminare oggetti. Tuttavia, si vuole limitare l'accesso di ciascun utente a un unico prefisso (cartella) nel bucket. Potresti creare cartelle con nomi che corrispondono ai loro nomi utente.

```
example-s3-bucket1
   Mary/
   Carlos/
```

Per assegnare a ciascun utente unicamente l'accesso alla cartella, è possibile scrivere una policy per ogni utente e collegarla singolarmente. È ad esempio possibile collegare la seguente policy all'utente Mary per concederle autorizzazioni Amazon S3 specifiche sulla cartella *example-s3-bucket1*/*Mary*.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket1/Mary/*"
      }
   ]
}
```

Successivamente, è possibile collegare una policy simile all'utente Carlos, specificando la cartella *Carlos* nel valore `Resource`.

Invece di collegare le policy ai singoli utenti, è possibile scrivere un'unica policy che utilizzi una variabile di policy, collegandola poi a un gruppo. In primo luogo, occorre creare un gruppo e aggiungervi Mary e Carlos. La seguente policy di esempio concede un set di autorizzazioni Amazon S3 nella cartella *example-s3-bucket1*/${aws:username}. Quando la politica viene valutata, la variabile di policy ${aws:username} viene sostituita dal nome utente del richiedente. Ad esempio, se Mary invia una richiesta di inserimento di un oggetto, l'operazione è consentita solo se l'oggetto viene caricato nella cartella *example-s3-bucket1*/Mary.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket1/${aws:username}/*"
      }
   ]
}
```

> **Note**
>
> Quando si utilizzano le variabili di policy, è necessario specificare esplicitamente la versione 2012-10-17 nella policy. La versione di default del linguaggio della policy IAM, 2008-10-17, non supporta le variabili di policy.

Se si vuole testare la policy precedente nella console di Amazon S3, la console deve avere l'autorizzazione per ottenere autorizzazioni aggiuntive, come illustrato nella policy seguente. Per ulteriori informazioni su come la console utilizza queste autorizzazioni, consulta Procedura guidata: controllo dell'accesso a un bucket con policy utente.

```
{
 "Version":"2012-10-17",
  "Statement": [
```

```
    {
      "Sid": "AllowGroupToSeeBucketListInTheConsole",
      "Action": [
       "s3:ListAllMyBuckets",
       "s3:GetBucketLocation"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Sid": "AllowRootLevelListingOfTheBucket",
      "Action": "s3:ListBucket",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::example-s3-bucket1",
      "Condition":{
            "StringEquals":{
                    "s3:prefix":[""], "s3:delimiter":["/"]
                           }
                }
    },
    {
      "Sid": "AllowListBucketOfASpecificUserPrefix",
      "Action": "s3:ListBucket",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::example-s3-bucket1",
      "Condition":{  "StringLike":{"s3:prefix":["${aws:username}/*"] }
       }
    },
      {
     "Sid": "AllowUserSpecificActionsOnlyInTheSpecificUserPrefix",
        "Effect":"Allow",
        "Action":[
           "s3:PutObject",
           "s3:GetObject",
           "s3:GetObjectVersion",
           "s3:DeleteObject",
           "s3:DeleteObjectVersion"
        ],
        "Resource":"arn:aws:s3:::example-s3-bucket1/${aws:username}/*"
      }
  ]
}
```

> **Note**
>
> Nella versione 2012-10-17 della policy, le variabili di policy iniziano con $. Questa modifica nella sintassi potenzialmente può creare un conflitto se la chiave dell'oggetto (nome dell'oggetto) include un $.
> Per evitare questo conflitto, specificare il carattere $ usando `${$}`. Ad esempio, per includere una chiave dell'oggetto `my$file` in una policy, si specifica il carattere con `my${$}file`.

Sebbene i nomi utente IAM siano identificatori semplici, in formato leggibile, non sono necessariamente univoci a livello globale. Ad esempio, se l'utente Carlos lascia l'organizzazione ed entra un altro utente Carlos, il Carlos nuovo potrebbe accedere alle informazioni del precedente Carlos.

Invece di utilizzare i nomi utente, puoi creare cartelle basate sugli ID utente IAM. Ogni ID utente IAM è univoco. In questo caso, si deve modificare la policy precedente per utilizzare la variabile di policy `${aws:userid}`. Per ulteriori informazioni sugli identificatori utente, consulta Identificatori IAMnella Guida per l'utente di IAM.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket1/home/${aws:userid}/*"
      }
   ]
}
```

Concessione a utenti non IAM (utenti dell'app per dispositivi mobili) dell'accesso alle cartelle in un bucket

Si supponga che si vuole sviluppare un'app mobile, un gioco che archivia i dati degli utenti in un bucket S3. Per ogni utente dell'app, si vuole creare una cartella nel bucket. Inoltre, desideri limitare l'accesso di ogni utente alla propria cartella. Ma non puoi creare cartelle prima che qualcuno scarichi la tua app e inizi a giocare, perché non hai il suo ID utente.

In questo caso, è possibile richiedere agli utenti di accedere all'app mediante provider di identità pubblici quali Login with Amazon, Facebook o Google. Una volta che gli utenti hanno effettuato l'accesso all'app mediante uno di questi provider, dispongono di un ID utente che può essere utilizzato per creare cartelle specifiche di un determinato utente al runtime.

Puoi quindi utilizzare la federazione delle identità web AWS Security Token Service per integrare le informazioni del provider di identità con la tua app e ottenere credenziali di sicurezza temporanee per ogni utente. Successivamente è possibile creare policy IAM che permettono all'app di accedere al bucket ed eseguire operazioni come la creazione di cartelle specifiche dell'utente e il caricamento dei dati. Per ulteriori informazioni sulla federazione delle identità Web, consulta Informazioni sulla federazione delle identità Web nella Guida per l'utente di IAM.

Concessione a un gruppo dell'accesso a una cartella condivisa in Amazon S3

Collegando la policy seguente al gruppo viene concesso a tutti i membri del gruppo l'accesso alla seguente cartella in Amazon S3: *example-s3-bucket1*/share/marketing. I membri del gruppo possono accedere solo alle autorizzazioni Amazon S3 specifiche illustrate nella policy e unicamente per gli oggetti nella cartella specificata.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket1/share/marketing/*"
      }
```

```
   ]
}
```

Permesso a tutti gli utenti di leggere gli oggetti in una parte del bucket

In questo esempio viene creato un gruppo denominato *AllUsers*, che contiene tutti gli utenti IAM che appartengono all' Account AWS. Viene collegata quindi una policy che consente al gruppo di accedere a `GetObject` e `GetObjectVersion`, ma solo per gli oggetti nella cartella *example-s3-bucket1*`/readonly`.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket1/readonly/*"
      }
   ]
}
```

Permesso a un partner di rilasciare i file in una parte specifica del bucket

In questo esempio, viene creato un gruppo denominato *AnyCompany* che rappresenta un'azienda partner. Viene creato un utente IAM per la persona o l'applicazione specifica presso l'azienda partner che ha necessità di effettuare l'accesso, quindi l'utente viene inserito nel gruppo.

Viene collegata quindi una policy che consente al gruppo di accedere alla cartella seguente in un bucket aziendale:

example-s3-bucket1`/uploads/anycompany`

Desideri inoltre impedire al gruppo *AnyCompany* di eseguire qualsiasi altra operazione con il bucket, quindi aggiungi un'istruzione che rifiuta esplicitamente l'autorizzazione per qualsiasi azione Amazon S3 ad eccezione di `PutObject` su qualsiasi risorsa Amazon S3 nell'account Account AWS.

```
{
```

```
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"s3:PutObject",
         "Resource":"arn:aws:s3:::example-s3-bucket1/uploads/anycompany/*"
      },
      {
         "Effect":"Deny",
         "Action":"s3:*",
         "NotResource":"arn:aws:s3:::example-s3-bucket1/uploads/anycompany/*"
      }
   ]
}
```

Restrizione dell'accesso ai bucket Amazon S3 in un Account AWS specifico

Se vuoi assicurarti che i tuoi responsabili di Amazon S3 accedano solo alle risorse che si trovano all'interno di un account affidabile Account AWS, puoi limitare l'accesso. Ad esempio, questa Policy IAM basata sull'identità utilizza un effetto `Deny` per bloccare l'accesso alle azioni Amazon S3, a meno che la risorsa Amazon S3 a cui si accede non sia presente nell'account *222222222222*. Per impedire a un principale IAM di accedere a oggetti Amazon S3 al di fuori dell'account, allega la seguente policy IAM: Account AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyS3AccessOutsideMyBoundary",
      "Effect": "Deny",
      "Action": [
        "s3:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": [
            "222222222222"
          ]
        }
      }
    }
  ]
```

```
}
```

> **Note**
>
> Questa policy non sostituisce i controlli di accesso IAM esistenti, perché non concede alcun accesso. Invece, questa policy funge da guardrail aggiuntivo per le altre autorizzazioni IAM, indipendentemente dalle autorizzazioni concesse tramite altre policy IAM.

Assicurati di sostituire l'ID account *222222222222* nella policy con il tuo Account AWS. Per applicare una policy a più account pur mantenendo questa restrizione, sostituire l'ID account con la chiave di condizione `aws:PrincipalAccount`. Questa condizione richiede che il principale e la risorsa devono trovarsi nello stesso account.

Limitazione dell'accesso ai bucket Amazon S3 all'interno dell'unità organizzativa

Se disponi di un'unità organizzativa (OU) configurata in AWS Organizations, potresti voler limitare l'accesso ai bucket Amazon S3 a una parte specifica dell'organizzazione. In questo esempio, utilizziamo la chiave `aws:ResourceOrgPaths` per limitare l'accesso del bucket Amazon S3 a un'unità organizzativa della tua organizzazione. In questo esempio, l'ID dell'unità organizzativa è *ou-acroot-exampleou*. Assicurati di sostituire questo valore nella tua policy con i tuoi ID dell'unità organizzativa.

```
{
   "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**
>
> Questa politica non concede alcun accesso. Al contrario, questa policy funge da backstop per le altre autorizzazioni IAM, impedendo ai tuoi principali di accedere a oggetti Amazon S3 al di fuori di un limite definito dall'unità organizzativa.

La policy nega l'accesso alle azioni di Amazon S3 a meno che l'oggetto Amazon S3 a cui si accede non si trovi nell'UO *`ou-acroot-exampleou`* nella tua organizzazione. La Condizione di policy IAM richiede `aws:ResourceOrgPaths`, una chiave di condizione multivalore, per contenere uno qualsiasi dei percorsi dell'unità organizzativa elencati. La politica utilizza l'operatore `ForAllValues:StringNotLike` per confrontare i valori di `aws:ResourceOrgPaths` alle UO elencate senza corrispondenza in base a maiuscole e minuscole.

Limitazione dell'accesso ai bucket Amazon S3 all'interno dell'organizzazione

Per limitare l'accesso agli oggetti Amazon S3 all'interno dell'organizzazione, allega una policy IAM alla radice dell'organizzazione, applicandola a tutti gli account dell'organizzazione. Per richiedere ai tuoi principale IAM di seguire questa regola, usa una policy di controllo dei servizi (SCP). Se scegli di utilizzare una SCP, assicurati di testare l'SCP prima di allegare la policy alla radice dell'organizzazione.

Nella seguente policy di esempio, l'accesso viene negato alle azioni di Amazon S3 a meno che l'oggetto Amazon S3 a cui si accede non si trovi nella stessa organizzazione del principale IAM che vi sta accedendo:

```
{
   "Version": "2012-10-17",
   "Statement": [
     {
       "Sid": "DenyS3AccessOutsideMyBoundary",
       "Effect": "Deny",
       "Action": [
         "s3:*"
       ],
       "Resource": "arn:aws:s3:::*/*",
       "Condition": {
```

```
        "StringNotEquals": {
          "aws:ResourceOrgID": "${aws:PrincipalOrgID}"
        }
      }
    }
  ]
}
```

> **Note**
>
> Questa politica non concede alcun accesso. Invece, questa policy funge da backstop per le altre autorizzazioni IAM, impedendo ai tuoi principali di accedere a qualsiasi oggetto Amazon S3 al di fuori della tua organizzazione. Questa policy si applica anche alle risorse Amazon S3 create dopo l'entrata in vigore della policy.

La Condizione di policy IAM in questo esempio richiede che `aws:ResourceOrgID` e `aws:PrincipalOrgID` siano uguali l'uno all'altro. Con questo requisito, il principale che effettua la richiesta e la risorsa a cui si accede devono trovarsi nella stessa organizzazione.

Concessione del permesso di recuperare la PublicAccessBlock configurazione per un Account AWS

L'esempio seguente di politica basata sull'identità concede l'autorizzazione a un utente. `s3:GetAccountPublicAccessBlock` Per queste autorizzazioni, è necessario impostare il valore `Resource` su `"*"`. Per informazioni sugli ARN delle risorse, vedere. Risorse relative alle policy per Amazon S3

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action":[
            "s3:GetAccountPublicAccessBlock"
         ],
         "Resource":[
            "*"
         ]
       }
    ]
```

```
}
```

Limitazione della creazione di bucket a una sola regione

Supponiamo che un Account AWS amministratore voglia concedere al proprio utente (Dave) l'autorizzazione a creare un bucket solo nella regione del Sud America (San Paolo). L'amministratore dell'account può collegare la seguente policy utente assegnando l'autorizzazione `s3:CreateBucket` con una condizione, come mostrato. La coppia chiave-valore nel blocco `Condition` specifica la chiave `s3:LocationConstraint` e la regione `sa-east-1` come valore corrispondente.

> **Note**
>
> In questo esempio, il proprietario di bucket sta assegnando un'autorizzazione a uno dei suoi utenti, in modo da poter utilizzare una policy di bucket o una policy utente. Questo esempio mostra una policy utente.

Per un elenco delle regioni di Amazon S3, consultare la sezione relativa a regioni ed endpoint nella Riferimenti generali di AWS.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action": "s3:CreateBucket",
         "Resource": "arn:aws:s3:::*",
         "Condition": {
             "StringLike": {
                 "s3:LocationConstraint": "sa-east-1"
             }
         }
       }
    ]
}
```

Aggiunta del rifiuto esplicito

La policy precedente limita l'utente impedendogli di creare un bucket in qualsiasi altra regione al di fuori di `sa-east-1`. Tuttavia, qualche altra policy potrebbe assegnare a questo utente

l'autorizzazione a creare bucket in un'altra regione. Ad esempio, se l'utente appartiene a un gruppo, è possibile che questo gruppo abbia una policy collegata che assegna a tutti gli utenti del gruppo stesso l'autorizzazione a creare bucket in un'altra regione. Per garantire che l'utente non ottenga l'autorizzazione a creare bucket in nessun'altra regione, puoi aggiungere una dichiarazione di rifiuto esplicita nella politica di cui sopra.

L'istruzione `Deny` utilizza la condizione `StringNotLike`. Cioè, la richiesta di creazione del bucket viene rifiutata se il vincolo di posizione non è `sa-east-1`. La negazione esplicita non consente all'utente di creare un bucket in nessun'altra regione, indipendentemente dalle altre autorizzazioni ottenute dall'utente. La seguente politica include una dichiarazione di negazione esplicita.

```
{
   "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"
             }
         }
       }
    ]
}
```

Verifica la politica con il AWS CLI

È possibile testare la politica utilizzando il seguente `create-bucket` AWS CLI comando. Questo esempio utilizza il file `bucketconfig.txt` per specificare il vincolo di posizione. Annotate il percorso Windows del file. È necessario aggiornare il nome del bucket e il percorso come opportuno. È necessario fornire le credenziali utente utilizzando il parametro `--profile`. Per ulteriori informazioni sulla configurazione e l'utilizzo di AWS CLI, vedereSviluppo con Amazon S3 tramite la AWS CLI.

```
aws s3api create-bucket --bucket examplebucket --profile AccountADave --create-bucket-
configuration file://c:/Users/someUser/bucketconfig.txt
```

Il file `bucketconfig.txt` specifica la configurazione come segue:

```
{"LocationConstraint": "sa-east-1"}
```

Procedura guidata: controllo dell'accesso a un bucket con policy utente

Questa spiegazione passo per passo illustra il funzionamento delle autorizzazioni utente con Amazon S3. In questo esempio, viene creato un bucket con cartelle. Quindi crei utenti AWS Identity and Access Management IAM nel tuo Account AWS e concedi a tali utenti autorizzazioni incrementali sul tuo bucket Amazon S3 e sulle cartelle in esso contenute.

Argomenti

Principi di base relativi a bucket e cartelle

Il modello di dati di Amazon S3 è una struttura flat: crei un bucket e il bucket archivia gli oggetti. Non c'è nessuna gerarchia di bucket secondari o sottocartelle, ma è possibile emulare una gerarchia delle cartelle. Strumenti come la console di Amazon S3 possono presentare una panoramica di queste cartelle e sottocartelle logiche nel bucket.

La console mostra che un bucket denominato `companybucket` ha tre cartelle, `Private`, `Development` e `Finance` e un oggetto, `s3-dg.pdf`. La console utilizza i nomi oggetto (chiavi) per creare una gerarchia logica con cartelle e sottocartelle. Considerare i seguenti esempi:

- Quando crei la cartella `Development`, la console crea un oggetto con la chiave `Development/`. Nota il delimitatore finale '/' (`/`).
- Quando carichi un oggetto denominato `Projects1.xls` nella cartella `Development`, la console carica l'oggetto e gli assegna la chiave `Development/Projects1.xls`.

  Nella chiave, `Development` è il prefisso e `/` è il delimitatore. L'API Amazon S3 supporta prefissi e delimitatori nelle operazioni. Ad esempio, è possibile ottenere un elenco di tutti gli oggetti da un bucket con un prefisso e un delimitatore specifici. Nella console, quando apri la cartella `Development`, viene visualizzato un elenco degli oggetti in essa contenuti. Nell'esempio seguente, la cartella `Development` contiene un solo oggetto.

  Quando la console visualizza la cartella `Development` nel bucket `companybucket`, invia una richiesta ad Amazon S3 in cui specifica un prefisso `Development` e un delimitatore `/`. La risposta della console si presenta proprio come un elenco di cartelle nel file system del computer. L'esempio precedente mostra che il bucket `companybucket` ha un oggetto con la chiave `Development/Projects1.xls`.

La console utilizza le chiavi degli oggetti per dedurre una gerarchia logica. Amazon S3 non ha una gerarchia fisica. Amazon S3 dispone solo di bucket che contengono oggetti in una struttura di file piatta. Quando si creano oggetti utilizzando l'API Amazon S3, è possibile utilizzare le chiavi degli oggetti che implicano una gerarchia logica. Quando viene creata una gerarchia logica di oggetti, è possibile gestire l'accesso alle singole cartelle, come dimostrato in questa procedura guidata.

Prima di iniziare, assicurarsi di acquisire familiarità con il concetto di contenuto di un bucket a livello di root. Si supponga che il bucket `companybucket` abbia i seguenti oggetti:

- `Private/privDoc1.txt`

- `Private/privDoc2.zip`
- `Development/project1.xls`
- `Development/project2.xls`
- `Finance/Tax2011/document1.pdf`
- `Finance/Tax2011/document2.pdf`
- `s3-dg.pdf`

Queste chiavi degli oggetti creano una gerarchia logica con `Private`, `Development` e `Finance` come cartelle a livello root e `s3-dg.pdf` come un oggetto a livello root. Quando si sceglie il nome di un bucket nella console di Amazon S3, le voci a livello di root vengono visualizzate. La console mostra i prefissi di livello superiore (`Private/`, `Development/` e `Finance/`) come cartelle a livello di root. La chiave dell'oggetto `s3-dg.pdf` non ha prefisso e quindi appare come voce a livello root.

Riepilogo della spiegazione passo per passo

In questa procedura guidata, creare un bucket con tre cartelle (`Private`, `Development` e `Finance`) al suo interno.

Ci sono due utenti, Alice e Bob. Alice deve accedere solo alla cartella `Development`, mentre Bob deve accedere solo alla cartella `Finance`. Il contenuto della cartella `Private` deve essere mantenuto privato. Nella procedura dettagliata, gestisci l'accesso creando utenti IAM (l'esempio utilizza i nomi utente Alice e Bob) e concedendo loro le autorizzazioni necessarie.

IAM supporta inoltre la creazione di gruppi di utenti e la concessione di autorizzazioni a livello di gruppo valide per tutti gli utenti presenti nel gruppo. In questo modo è possibile gestire le autorizzazioni in modo più efficiente. Per questo esercizio sia Alice che Bob hanno bisogno di autorizzazioni comuni. Pertanto, verrà creato anche un gruppo denominato `Consultants` e Alice e Bob saranno aggiunti al gruppo. Inizialmente, le autorizzazioni vengono assegnate collegando una policy di gruppo al gruppo stesso. Quindi, vengono aggiunte autorizzazioni specifiche per gli utenti collegando le policy agli utenti specifici.

> **Note**
>
> La spiegazione passo per passo utilizza `companybucket` come nome del bucket, Alice e Bob come utenti IAM e `Consultants` come nome del gruppo. Poiché Amazon S3 richiede

che i nomi di bucket siano univoci a livello globale, è necessario sostituire il nome del bucket con un nome personalizzato.

Preparazione della procedura guidata

In questo esempio, utilizzi le tue Account AWS credenziali per creare utenti IAM. Inizialmente, questi utenti non hanno autorizzazioni. Le autorizzazioni vengono concesse in modo incrementale per l'esecuzione di operazioni di Amazon S3 specifiche. Per testare queste autorizzazioni, viene effettuato l'accesso alla console con le credenziali di ciascun utente. Man mano che concedi in modo incrementale le autorizzazioni come Account AWS proprietario e le testi come utente IAM, devi accedere e disconnetterti, ogni volta utilizzando credenziali diverse. È possibile eseguire questo test con un browser, ma il processo sarà più rapido se è possibile utilizzare due browser diversi. Utilizza un browser per connetterti a AWS Management Console con le tue Account AWS credenziali e un altro browser per connetterti con le credenziali utente IAM.

Per accedere al AWS Management Console con le tue Account AWS credenziali, vai su https://console.aws.amazon.com/. Un utente IAM non può accedere utilizzando lo stesso link. Un utente IAM deve utilizzare una pagina di accesso abilitata per IAM. Come proprietario dell'account, è possibile fornire questo link agli utenti.

Per ulteriori informazioni su IAM, consulta la pagina di accesso alla AWS Management Console nella Guida per l'utente di IAM.

Per fornire un collegamento di accesso agli utenti IAM

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro Navigation (Navigazione) sceglIAM Dashboard (Pannello di controllo IAM).
3. Prendere nota dell'URL in IAM users sign in link (Collegamento di accesso utenti IAM). Sarà necessario fornire questo collegamento agli utenti IAM affinché possano accedere alla console con il loro nome utente e la loro password IAM.

Fase 1: creazione di un bucket

In questo passaggio, accedi alla console Amazon S3 con Account AWS le tue credenziali, crei un bucket, aggiungi cartelle al bucket e carichi uno o due documenti di esempio in ogni cartella.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Creare un bucket.

   Per step-by-step istruzioni, consultaCreazione di un bucket.
3. Caricare un documento nel bucket.

   Questo esercizio presume che il documento `s3-dg.pdf` si trovi a livello root di questo bucket. Se viene caricato un documento differente, è necessario sostituire il nome file con `s3-dg.pdf`.
4. Aggiungere tre cartelle denominate `Private`, `Finance` e `Development` al bucket.

   Per step-by-step istruzioni su come creare una cartella, consulta Organizzazione degli oggetti nella console di Amazon S3 utilizzando le cartelle > nella Guida per l'utente di Amazon Simple Storage Service.
5. Caricare uno o due documenti in ciascuna cartella.

   Per questo esercizio, si presume che siano stati caricati un paio di documenti in ciascuna cartella, in modo che il bucket abbia oggetti con le seguenti chiavi:

   - `Private/privDoc1.txt`
   - `Private/privDoc2.zip`
   - `Development/project1.xls`
   - `Development/project2.xls`
   - `Finance/Tax2011/document1.pdf`
   - `Finance/Tax2011/document2.pdf`
   - `s3-dg.pdf`

   Per step-by-step istruzioni, consultaCaricamento degli oggetti.

Fase 2: creazione di un gruppo e di utenti IAM

Ora usa la console IAM per aggiungere due utenti IAM, Alice e Bob, al tuo Account AWS. Per step-by-step istruzioni, consulta Creating an IAM user in your Account AWS nella IAM User Guide.

Crea anche un gruppo amministrativo denominato`Consultants`. Quindi aggiungi entrambi gli utenti al gruppo. Per step-by-step istruzioni, consulta Creazione di gruppi di utenti IAM.

> **Warning**
>
> Quando si aggiungono utenti e un gruppo, non collegare alcuna policy che assegni autorizzazioni agli utenti. Inizialmente, questi utenti non dispongono di alcuna autorizzazione. Nelle sezioni seguenti, le autorizzazioni vengono concesse in modo incrementale. In primo luogo, devi accertarti di avere assegnato le password a questi utenti IAM. Queste credenziali utente vengono utilizzate per testare le operazioni di Amazon S3 e verificare che le autorizzazioni funzionino come previsto.

Per step-by-step istruzioni sulla creazione di un nuovo utente IAM, consulta Creating an IAM user in your Account AWS nella IAM User Guide. Quando crei gli utenti per questa procedura guidata, seleziona Accesso alla AWS Management Console e deseleziona Accesso programmatico.

Per step-by-step istruzioni sulla creazione di un gruppo amministrativo, consulta Creating Your First IAM Admin User and Group nella IAM User Guide.

Fase 3: verifica che gli utenti IAM non dispongano di autorizzazioni

Se stai utilizzando due browser, puoi ora utilizzare il secondo browser per effettuare l'accesso alla console con una delle credenziali utente IAM.

1. Utilizzando il link di accesso dell'utente IAM (consulta Per fornire un collegamento di accesso agli utenti IAM), effettua l'accesso alla AWS Management Console utilizzando una delle credenziali utente IAM.
2. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.

   Verifica il messaggio della console che ti informa che l'accesso è negato.

Ora, è possibile iniziare a concedere le autorizzazioni incrementali agli utenti. Innanzitutto, collegare una policy di gruppo che concede le autorizzazioni necessarie per entrambi gli utenti.

Fase 4: concessione di autorizzazioni a livello di gruppo

Gli utenti devono essere in grado di effettuare quanto segue:

- Elencare tutti i bucket di proprietà dell'account padre. A tale scopo, Bob e Alice devono avere l'autorizzazione per l'operazione `s3:ListAllMyBuckets`.

- Elencare le voci, le cartelle e gli oggetti a livello root nel bucket `companybucket`. A tale scopo, Bob e Alice devono avere l'autorizzazione per l'operazione `s3:ListBucket` nel bucket `companybucket`.

Innanzitutto, creare una policy che concede tali autorizzazioni e quindi collegarla al gruppo `Consultants`.

Fase 4.1: concessione di autorizzazione per elencare tutti i bucket

In questa fase viene creata una policy gestita che concede agli utenti le autorizzazioni minime per consentire loro di elencare tutti i bucket di proprietà dell'account padre. Quindi, tale policy verrà collegata al gruppo `Consultants`. Quando si collega la policy gestita a un utente o a un gruppo, si concede all'utente o al gruppo l'autorizzazione per ottenere un elenco dei bucket di proprietà dell' Account AWS parent.

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

   > **Note**
   >
   > Poiché stai concedendo autorizzazioni utente, è necessario effettuare l'accesso con le credenziali dell' Account AWS , non come utente IAM.

2. Creare la policy gestita.

   a. Nel riquadro di navigazione sulla sinistra, selezionare Policies (Policy) e scegliere Create Policy (Crea policy).

   b. Selezionare la scheda JSON.

   c. Copiare la policy di accesso seguente e incollarla nel campo di testo relativo alla policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowGroupToSeeBucketListInTheConsole",
      "Action": ["s3:ListAllMyBuckets"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::*"]
    }
```

```
  ]
}
```

Una policy è un documento JSON. Nel documento, uno `Statement` è una serie di oggetti, ognuno dei quali descrive un'autorizzazione utilizzando un insieme di coppie di nome-valore. La suddetta policy descrive un'autorizzazione specifica. L'`Action` specifica il tipo di accesso. Nella policy, `s3:ListAllMyBuckets` è un'operazione di Amazon S3 predefinita. Questa azione riguarda l'operazione del servizio Amazon S3 GET, che restituisce un elenco di tutti i bucket di proprietà del mittente autenticato. Il valore dell'elemento `Effect` determina se un'autorizzazione specifica è consentita o rifiutata.

d. Scegliere Review policy (Esamina policy). Nella pagina successiva, immettere `AllowGroupToSeeBucketListInTheConsole` nel campo Name (Nome), quindi scegliere Create policy (Crea policy).

> **Note**
>
> La voce Summary (Riepilogo) visualizza un messaggio in cui si afferma che la policy non concede alcuna autorizzazione. Per questa procedura guidata, il messaggio può essere ignorato.

3. Collegare la policy gestita `AllowGroupToSeeBucketListInTheConsole` che è stata creata al gruppo `Consultants`.

   Per step-by-step istruzioni su come allegare una policy gestita, consulta Aggiungere e rimuovere le autorizzazioni di identità IAM nella Guida per l'utente IAM.

   I documenti della policy vengono collegati agli utenti e ai gruppi IAM nella console IAM. Poiché entrambi gli utenti devono essere in grado di elencare i bucket, la policy deve essere collegata al gruppo.

4. Testare l'autorizzazione.

   a. Utilizzando il collegamento di accesso utente IAM (consultare Per fornire un collegamento di accesso agli utenti IAM), accedere alla console con una delle credenziali utente IAM.

   b. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.

      La console ora dovrebbe elencare tutti i bucket, ma non gli oggetti contenuti in ogni bucket.

Fase 4.2: abilitazione degli utenti a elencare il contenuto di un bucket a livello root

Di seguito, consentire a tutti gli utenti nel gruppo `Consultants` di elencare le voci del bucket `companybucket` a livello root. Quando un utente sceglie il bucket aziendale nella console di Amazon S3, può visualizzare le voci a livello root nel bucket.

> **Note**
>
> Questo esempio utilizza `companybucket` a scopo illustrativo. È necessario utilizzare il nome del bucket che è stato creato.

Per comprendere la richiesta che la console invia ad Amazon S3 quando scegli il nome di un bucket, la risposta che Amazon S3 restituisce e come la console interpreta la risposta, esamina il flusso un po' più da vicino.

Quando viene scelto un nome di bucket, la console invia la richiesta GET Bucket (ListObjects) ad Amazon S3. Questa richiesta include i seguenti parametri:

- Il parametro `prefix` che presenta una stringa vuota come valore.
- Il `delimiter` parametro con `/` come valore.

Di seguito è riportata una richiesta di esempio.

```
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 restituisce una risposta che include il seguente 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>
```

L'oggetto `s3-dg.pdf` della chiave non contiene il delimitatore barra (`/`) e Amazon S3 restituisce la chiave nell'elemento `<Contents>`. Tutte le altre chiavi nel bucket di esempio contengono tuttavia il delimitatore `/`. Amazon S3 raggruppa queste chiavi e restituisce un elemento `<CommonPrefixes>` per ciascuno dei diversi valori di prefisso `Development/`, `Finance/` e `Private/` che corrisponde a una sottostringa dall'inizio di queste chiavi alla prima occorrenza del delimitatore `/` specificato.

La console interpreta questo risultato e mostra le voci a livello root come tre cartelle e una chiave dell'oggetto.

Se Bob o Alice apre la cartella Development, la console invia la richiesta GET Bucket (ListObjects) ad Amazon S3 con i parametri `prefix` e `delimiter` impostati sui seguenti valori:

- Il parametro `prefix` con il valore `Development/`.
- Il parametro `delimiter` con il valore "`/`".

In risposta, Amazon S3 restituisce le chiavi degli oggetti che iniziano con il prefisso specificato.

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>companybucket</Name>
  <Prefix>Development</Prefix>
  <Delimiter>/</Delimiter>
   ...
  <Contents>
    <Key>Project1.xls</Key>
    ...
  </Contents>
  <Contents>
    <Key>Project2.xls</Key>
    ...
  </Contents>
```

```
</ListBucketResult>
```

La console mostra le chiavi degli oggetti.

Ora, tornare alla concessione dell'autorizzazione agli utenti per elencare le voci del bucket a livello root. Per elencare il contenuto del bucket, gli utenti devono disporre dell'autorizzazione per chiamare l'operazione `s3:ListBucket`, come illustrato nella seguente dichiarazione di policy. Per fare in modo che possa essere visualizzato il contenuto a livello root, è necessario aggiungere una condizione per richiedere che gli utenti specifichino un oggetto `prefix` vuoto nella richiesta, ovvero gli utenti non sono autorizzati a fare doppio clic su alcuna cartella a livello root. Infine, aggiungere una condizione per esigere un accesso di tipo cartella imponendo che le richieste dell'utente includano il parametro `delimiter` con il valore "`/`".

```
{
  "Sid": "AllowRootLevelListingOfCompanyBucket",
  "Action": ["s3:ListBucket"],
  "Effect": "Allow",
  "Resource": ["arn:aws:s3:::companybucket"],
  "Condition":{
         "StringEquals":{
             "s3:prefix":[""], "s3:delimiter":["/"]
                       }
              }
}
```

Quando scegli un bucket sulla console Amazon S3, la console invia innanzitutto la richiesta di posizione GET Bucket per trovare dove è distribuito Regione AWS il bucket. La console utilizza quindi l'endpoint specifico della regione per il bucket per inviare la richiesta GET Bucket (ListObjects). Di conseguenza, se gli utenti utilizzeranno la console, è necessario assegnare l'autorizzazione per l'operazione `s3:GetBucketLocation` come illustrato nella seguente dichiarazione di policy.

```
{
   "Sid": "RequiredByS3Console",
   "Action": ["s3:GetBucketLocation"],
   "Effect": "Allow",
   "Resource": ["arn:aws:s3:::*"]
}
```

Per consentire agli utenti di elencare il contenuto di un bucket a livello root

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

   Usa Account AWS le tue credenziali, non le credenziali di un utente IAM, per accedere alla console.

2. Sostituire la policy gestita `AllowGroupToSeeBucketListInTheConsole` esistente che è collegata al gruppo `Consultants` con la seguente policy, che consente anche l'operazione `s3:ListBucket`. Ricordati di sostituirlo *`companybucket`*nella policy `Resource` con il nome del tuo bucket.

   Per step-by-step istruzioni, consulta Modifica delle politiche IAM nella Guida per l'utente IAM. Quando segui le step-by-step istruzioni, assicurati di seguire i passaggi per applicare le modifiche a tutte le principali entità a cui è allegata la policy.

```
{
  "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. Test delle autorizzazioni aggiornate.

a. Mediante il link di accesso dell'utente IAM (consulta Per fornire un collegamento di accesso agli utenti IAM), accedere alla AWS Management Console.

   Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.

b. Scegliere il bucket creato. La console mostra le voci del bucket a livello root. Se si sceglie qualsiasi cartella nel bucket, non sarà possibile visualizzare il contenuto della cartella perché le relative autorizzazioni non sono state ancora concesse.

Questo test ha esito positivo quando gli utenti utilizzano la console di Amazon S3. Quando si sceglie un bucket sulla console, l'implementazione della console invia una richiesta che include il parametro `prefix` con una stringa vuota come valore e il parametro `delimiter` con "/" come valore.

Fase 4.3: sintesi della policy di gruppo

L'effetto della policy di gruppo aggiunta è quello di concedere agli utenti IAM Alice e Bob le seguenti autorizzazioni minime:

- Elencare tutti i bucket di proprietà dell'account padre.
- Visualizzare le voci a livello root nel bucket `companybucket`.

Tuttavia, gli utenti ancora non possono fare molto. Di seguito, concedere autorizzazioni specifiche per utente, come segue:

- Consentire a Alice di prendere e mettere oggetti nella cartella `Development`.
- Consentite a Bob di prendere e mettere oggetti nella cartella `Finance`.

Per le autorizzazioni specifiche dell'utente, collegare una policy all'utente specifico, non al gruppo. Nella sezione seguente, ad Alice vengono concesse le autorizzazioni per lavorare nella cartella `Development`. È possibile ripetere le fasi per concedere un'autorizzazione simile a Bob per lavorare nella cartella `Finance`.

Fase 5: concessione di autorizzazioni specifiche all'utente IAM Alice

È necessario ora concedere autorizzazioni aggiuntive ad Alice in modo che possa vedere il contenuto della cartella `Development` per poter prendere e mettere oggetti nella stessa.

Fase 5.1: concessione dell'autorizzazione a elencare il contenuto della cartella Development all'utente IAM Alice

Affinché Alice `Development` elenchi il contenuto della cartella, devi applicare all'utente Alice una politica che conceda l'autorizzazione per l'`s3:ListBucket`azione sul `companybucket` bucket, a condizione che la richiesta includa il prefisso. `Development/` Questa policy deve essere applicata solo all'utente Alice, pertanto viene utilizzata una policy inline. Per ulteriori informazioni sulle policy inline, consulta Policy gestite e policy inline nella Guida per l'utente di IAM.

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

   Usa Account AWS le tue credenziali, non le credenziali di un utente IAM, per accedere alla console.

2. Creare una policy inline per concedere all'utente Alice l'autorizzazione per elencare il contenuto della cartella `Development`.

   a. Nel riquadro di navigazione sinistro, scegliere Users (Utenti).

   b. Scegli il nome utente Alice.

   c. Nella pagina dei dettagli dell'utente, scegliere la scheda Permissions (Autorizzazioni), quindi selezionare Add inline policy (Aggiungi policy inline).

   d. Selezionare la scheda JSON.

   e. Copia la seguente politica e incollala nel campo di testo della politica.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
      "Sid": "AllowListBucketIfSpecificPrefixIsIncludedInRequest",
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::companybucket"],
      "Condition":{  "StringLike":{"s3:prefix":["Development/*"] }
       }
    }
  ]
}
```

f. Scegliere Review policy (Esamina policy). Nella pagina successiva, immettere un nome nel campo Name (Nome), quindi scegliere Create policy (Crea policy).

3. Test della modifica apportata alle autorizzazioni di Alice:

   a. Mediante il link di accesso dell'utente IAM (consulta Per fornire un collegamento di accesso agli utenti IAM), accedere alla AWS Management Console.

   b. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.

   c. Nella console di Amazon S3 verificare che Alice possa visualizzare l'elenco degli oggetti nella cartella `Development/` del bucket.

      Quando l'utente sceglie la cartella `/Development` per visualizzare l'elenco degli oggetti in essa contenuti, la console di Amazon S3 invia la richiesta `ListObjects` ad Amazon S3 con il prefisso `/Development`. Poiché all'utente è stata concessa l'autorizzazione per visualizzare l'elenco degli oggetti con il prefisso `Development` e il delimitatore `/`, Amazon S3 restituisce l'elenco degli oggetti con il prefisso della chiave `Development/` e la console visualizza tale elenco.

Fase 5.2: concessione delle autorizzazioni a recuperare e inserire oggetti nella cartella Development all'utente IAM Alice

Affinché Alice possa prendere e mettere oggetti nella cartella `Development`, ha bisogno di un'autorizzazione per chiamare le operazioni `s3:GetObject` e `s3:PutObject`. Le seguenti dichiarazioni di policy assegnano queste autorizzazioni purché la richiesta includa il parametro `prefix` con un valore di `Development/`.

```
{
    "Sid":"AllowUserToReadWriteObjectData",
    "Action":["s3:GetObject", "s3:PutObject"],
    "Effect":"Allow",
    "Resource":["arn:aws:s3:::companybucket/Development/*"]
 }
```

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

   Usa Account AWS le tue credenziali, non quelle di un utente IAM, per accedere alla console.

2. Modificare la policy inline creata nella fase precedente.

a. Nel riquadro di navigazione sinistro, scegliere Users (Utenti).

b. Scegliere il nome utente Alice.

c. Nella pagina dei dettagli, scegliere la scheda Permissions (Autorizzazioni) ed espandere la sezione Inline Policies (Policy inline).

d. Accanto al nome della policy creata nella fase precedente, scegliere Edit Policy (Modifica policy) .

e. Copiare la seguente policy e incollarla nel campo di testo della policy, sostituendo la policy esistente.

```
{
     "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. Test della policy aggiornata:

a. Mediante il link di accesso dell'utente IAM (consulta Per fornire un collegamento di accesso agli utenti IAM), accedere alla AWS Management Console.

b. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.

c. Nella console di Amazon S3 verificare che Alice possa aggiungere e scaricare un oggetto nella cartella `Development`.

Fase 5.3: rifiuto esplicito delle autorizzazioni relative a qualsiasi altra cartella del bucket per l'utente IAM Alice

L'utente Alice ora può elencare il contenuto del bucket `companybucket` a livello root. Inoltre, ora può prendere e mettere oggetti nella cartella `Development`. Se si vuole effettivamente limitare le autorizzazioni di accesso, è possibile rifiutare esplicitamente ad Alice l'accesso a qualsiasi altra cartella del bucket. Se esiste qualsiasi altra policy (policy di bucket o ACL) che assegna ad Alice l'accesso a eventuali altre cartelle del bucket, questo rifiuto esplicito sovrascrive tali autorizzazioni.

È possibile aggiungere la seguente istruzione alla policy utente di Alice, che prevede che tutte le richieste inviate da Alice ad Amazon S3 includano il parametro `prefix`, il cui valore può essere `Development/*` oppure una stringa vuota.

```
{
   "Sid": "ExplicitlyDenyAnyRequestsForAllOtherFoldersExceptDevelopment",
   "Action": ["s3:ListBucket"],
   "Effect": "Deny",
   "Resource": ["arn:aws:s3:::companybucket"],
   "Condition":{  "StringNotLike": {"s3:prefix":["Development/*",""] },
                  "Null"         : {"s3:prefix":false }
    }
}
```

Esistono due espressioni condizionali nel blocco `Condition`. Il risultato di queste espressioni condizionali viene combinato utilizzando l'`AND` logico. Se entrambe le condizioni sono vere, il risultato della condizione combinata è vero. Poiché `Effect` in questa policy è `Deny`, quando `Condition` viene valutata true, gli utenti non saranno in grado di eseguire la `Action` specificata.

- L'espressione condizionale `Null` assicura che le richieste provenienti da Alice includano il parametro `prefix`.

  Il parametro `prefix` richiede l'accesso di tipo cartella. Se viene inviata una richiesta senza il parametro `prefix`, Amazon S3 restituisce tutte le chiavi degli oggetti.

  Se la richiesta include il parametro `prefix` con un valore null, l'espressione restituisce il valore True, quindi tutta la `Condition` restituisce il valore True. È necessario consentire una stringa vuota come valore del parametro `prefix`. Da quanto detto in precedenza, ricordare che permettere una stringa nulla significa consentire ad Alice di recuperare le voci del bucket a livello

root come fa la console nella precedente discussione. Per ulteriori informazioni, consulta Fase 4.2: abilitazione degli utenti a elencare il contenuto di un bucket a livello root.

- L'espressione condizionale `StringNotLike` assicura che se il valore del parametro `prefix` viene specificato e non è `Development/*`, la richiesta ha esito negativo.

Seguire le fasi della sezione precedente e aggiornare nuovamente la policy inline creata per l'utente Alice.

Copiare la seguente policy e incollarla nel campo di testo della policy, sostituendo la policy esistente.

```
{
   "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 }
          }
      }
   ]
}
```

Fase 6: concessione di autorizzazioni specifiche all'utente IAM Bob

È necessario ora assegnare a Bob l'autorizzazione per la cartella `Finance`. Seguire le fasi utilizzate precedentemente per assegnare le autorizzazioni ad Alice ma sostituire la cartella `Development` con la cartella `Finance`. Per step-by-step istruzioni, consulta. Fase 5: concessione di autorizzazioni specifiche all'utente IAM Alice

Fase 7: protezione della cartella Private (Privato)

In questo esempio, vi sono soltanto due utenti. Sono state concesse le autorizzazioni minime a livello di gruppo e quelle a livello di utente unicamente quando erano veramente necessarie delle autorizzazioni a livello di singolo utente. Questo approccio contribuisce ad alleggerire l'impegno necessario per gestire le autorizzazioni. Con l'aumento del numero degli utenti, la gestione delle autorizzazioni può diventare gravosa. Ad esempio, non vogliamo che alcun utente di questo esempio acceda al contenuto della cartella `Private`. Come ci si assicura di non concedere accidentalmente a un utente l'autorizzazione alla `Private` cartella? È necessario aggiungere una policy che rifiuti esplicitamente l'accesso alla cartella. Un rifiuto esplicito sovrascrive qualsiasi altra autorizzazione.

Per essere certi che la cartella `Private` resti privata, è possibile aggiungere le seguenti due dichiarazioni di rifiuto alla policy di gruppo:

- Aggiungere la seguente dichiarazione per rifiutare esplicitamente qualsiasi operazione sulle risorse della cartella `Private` (`companybucket/Private/*`).

```
{
  "Sid": "ExplictDenyAccessToPrivateFolderToEveryoneInTheGroup",
  "Action": ["s3:*"],
  "Effect": "Deny",
  "Resource":["arn:aws:s3:::companybucket/Private/*"]
}
```

- Viene inoltre rifiutata l'autorizzazione a eseguire l'operazione di elenco degli oggetti quando la richiesta specifica il prefisso `Private/`. Nella console, se Bob o Alice apre la cartella `Private`, questa policy fa in modo che Amazon S3 restituisca una risposta di errore.

```
{
  "Sid": "DenyListBucketOnPrivateFolder",
  "Action": ["s3:ListBucket"],
  "Effect": "Deny",
  "Resource": ["arn:aws:s3:::*"],
  "Condition":{
```

```
      "StringLike":{"s3:prefix":["Private/"]}
    }
}
```

Sostituire la policy del gruppo `Consultants` con una policy aggiornata che includa le precedenti dichiarazioni di rifiuto. Una volta applicata la policy aggiornata, nessuno degli utenti del gruppo può accedere alla cartella `Private` nel bucket.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

   Usa Account AWS le tue credenziali, non quelle di un utente IAM, per accedere alla console.

2. Sostituire la policy gestita `AllowGroupToSeeBucketListInTheConsole` esistente che è collegata al gruppo `Consultants` con la seguente policy. È necessario ricordare di sostituire *`companybucket`* nella policy con il nome del bucket.

   Per istruzioni, consulta Modifica delle politiche gestite dai clienti nella Guida per l'utente IAM. Quando si seguono le istruzioni, osservare le indicazioni per l'applicazione delle modifiche a tutte le entità principali a cui è collegata la policy.

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

Fase 8: Pulizia

Per eseguire la pulizia, apri la console IAM e rimuovi gli utenti Alice e Bob. Per step-by-step istruzioni, consulta Eliminazione di un utente IAM nella Guida per l'utente IAM.

Per essere certi che non vengano addebitati costi aggiuntivi per lo storage, è necessario eliminare anche gli oggetti e il bucket che è stato creato per questo esercizio.

Risorse correlate

- Gestione di policy IAM nella Guida per l'utente IAM

Procedure dettagliate che utilizzano policy per gestire l'accesso alle risorse Amazon S3

Questo argomento contiene i seguenti esempi di procedure guidate introduttive per concedere l'accesso alle risorse di Amazon S3. Questi esempi li utilizzano AWS Management Console per creare risorse (bucket, oggetti, utenti) e concedere loro le autorizzazioni. Gli esempi mostrano quindi come verificare le autorizzazioni utilizzando gli strumenti a riga di comando per evitare di scrivere il codice. Forniamo comandi utilizzando sia il AWS Command Line Interface (AWS CLI) che il. AWS Tools for Windows PowerShell

- Esempio 1: il proprietario del bucket concede agli utenti le autorizzazioni per il bucket

  Per default, gli utenti IAM creati nell'account non dispongono delle autorizzazioni. In questo esercizio agli utenti verrà concessa un'autorizzazione per eseguire le operazioni sui bucket e sugli oggetti.

- Esempio 2: il proprietario del bucket concede autorizzazioni per il bucket multiaccount

  In questo esercizio un proprietario del bucket, Account A, concede le autorizzazioni multiaccount a un altro Account AWS, Account B, che delega quindi queste autorizzazioni agli utenti nel suo account.

- Gestione delle autorizzazioni per l'oggetto quando il proprietario dell'oggetto non corrisponde al proprietario del bucket

  Gli scenari di esempio in questo caso riguardano un proprietario del bucket che concede ad altri le autorizzazioni per l'oggetto, sebbene non tutti gli oggetti nel bucket siano di sua proprietà. Di quali autorizzazioni ha bisogno il proprietario del bucket e come può delegare tali autorizzazioni?

  Chi Account AWS crea un bucket si chiama proprietario del bucket. Il proprietario può concedere altre Account AWS autorizzazioni per caricare oggetti e chi crea Account AWS gli oggetti ne è proprietario. Il proprietario del bucket non dispone delle autorizzazioni per gli oggetti creati dagli altri Account AWS. Se il proprietario del bucket scrive una policy relativa al bucket che concede l'accesso agli oggetti, la policy non si applica agli oggetti di proprietà di altri account.

  In questo caso il proprietario dell'oggetto deve in primo luogo concedere le autorizzazioni al proprietario del bucket utilizzando un'ACL dell'oggetto. Il proprietario del bucket può quindi delegare le autorizzazioni relative agli oggetti ad altri, agli utenti del proprio account o a un altro Account AWS, come illustrato negli esempi seguenti.

- Esempio 3: il proprietario del bucket concede autorizzazioni per gli oggetti che non sono di sua proprietà

  In questo esercizio il proprietario del bucket ottiene prima le autorizzazioni dal proprietario dell'oggetto, Il proprietario del bucket quindi delega queste autorizzazioni agli utenti nel suo account.

- Esempio 4: il proprietario del bucket concede l'autorizzazione di più account a oggetti di cui non è proprietario

  Dopo aver ricevuto le autorizzazioni dal proprietario dell'oggetto, il proprietario del bucket non può delegare l'autorizzazione ad altri Account AWS perché la delega tra account non è supportata (vedi). Delega delle autorizzazioni Invece, il proprietario del bucket può creare un ruolo IAM con autorizzazioni per eseguire operazioni specifiche (come get object) e consentire a un altro di assumere quel ruolo. Account AWS Chiunque assuma il ruolo potrà quindi accedere agli oggetti. Questo esempio mostra in che modo un proprietario del bucket può utilizzare un ruolo IAM per abilitare questa delega multiaccount.

Prima di provare le procedure guidate di esempio

Questi esempi utilizzano il AWS Management Console per creare risorse e concedere autorizzazioni. Per verificare le autorizzazioni, gli esempi utilizzano gli strumenti della riga di comando e AWS CLI AWS Tools for Windows PowerShell, quindi, non è necessario scrivere alcun codice. Per testare le autorizzazioni, è necessario configurare uno di questi strumenti. Per ulteriori informazioni, consulta Configurazione degli strumenti per le procedure dettagliate.

Inoltre, durante la creazione di risorse, questi esempi non utilizzano le credenziali utente root di un. Account AWS ma viene creato un utente amministratore in questi account per eseguire queste attività.

Informazioni sull'uso di un utente amministratore per creare risorse e concedere autorizzazioni

AWS Identity and Access Management (IAM) sconsiglia di utilizzare le credenziali dell'utente root dell' Account AWS per effettuare richieste. Invece, crea un utente o un ruolo IAM, concedi a tale utente o ruolo l'accesso completo, quindi utilizza le relative credenziali per fare richieste. Questo utente viene definito utente o ruolo amministratore. Per ulteriori informazioni, consultare la sezione relativa a credenziali Utente root dell'account AWS e identità IAM nella Riferimenti generali di AWS e Best practice di IAM nella Guida per l'utente di IAM.

In tutte le procedure guidate di esempio riportate in questa sezione vengono utilizzate le credenziali dell'utente amministratore. Se non avete creato un utente amministratore per il vostro Account AWS, negli argomenti viene illustrato come fare.

Per accedere AWS Management Console utilizzando le credenziali utente, devi utilizzare l'URL di accesso utente IAM. La console IAM fornisce questo URL per il tuo. Account AWS Negli argomenti di questa sezione viene illustrato come ottenere l'URL.

Configurazione degli strumenti per le procedure dettagliate

Gli esempi introduttivi (vediProcedure dettagliate che utilizzano policy per gestire l'accesso alle risorse Amazon S3) utilizzano il AWS Management Console per creare risorse e concedere autorizzazioni. Per testare le autorizzazioni, gli esempi utilizzano gli strumenti della riga di comando, AWS Command Line Interface (AWS CLI) e AWS Tools for Windows PowerShell, quindi non è necessario scrivere alcun codice. Per testare le autorizzazioni, è necessario configurare uno di questi strumenti.

Per configurare AWS CLI

1. Scarica e configura la AWS CLI. Per le istruzioni, consulta i seguenti argomenti nella Guida per l'utente dell'AWS Command Line Interface :

   Installare o aggiornare alla versione più recente di AWS Command Line Interface

   Inizia con AWS Command Line Interface
2. Impostare il profilo di default.

   Le credenziali utente vengono memorizzate nel file di AWS CLI configurazione. Crea un profilo predefinito nel file di configurazione utilizzando le tue credenziali. Account AWS Per istruzioni su come trovare e modificare il file di AWS CLI configurazione, consulta Impostazioni del file di configurazione e credenziali.

   ```
   [default]
   aws_access_key_id = access key ID
   aws_secret_access_key = secret access key
   region = us-west-2
   ```
3. Verificare la configurazione digitando i comandi riportati di seguito al prompt dei comandi. Poiché entrambi questi comandi non forniscono credenziali in modo esplicito, vengono utilizzate le credenziali del profilo di default.

- Prova il comando. `help`

```
aws help
```

- Per ottenere un elenco di bucket sull'account configurato, usa il `aws s3 ls` comando.

```
aws s3 ls
```

Durante le procedure dettagliate, creerai gli utenti e salverai le credenziali degli utenti nei file di configurazione creando profili, come illustrato nell'esempio seguente. Questi profili hanno i nomi di e. `AccountAadmin` `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
```

Per eseguire un comando utilizzando queste credenziali utente, aggiungere il parametro `--profile` specificando il nome del profilo. Il AWS CLI comando seguente recupera un elenco di oggetti *`examplebucket`*e specifica il `AccountBadmin` profilo.

```
aws s3 ls s3://examplebucket --profile AccountBadmin
```

In alternativa, è possibile configurare un set di credenziali utente come profilo di default modificando la variabile di ambiente `AWS_DEFAULT_PROFILE` dal prompt dei comandi. Dopo aver eseguito questa operazione, ogni volta che si eseguono AWS CLI comandi senza il `--profile` parametro, AWS CLI utilizza il profilo impostato nella variabile di ambiente come profilo predefinito.

```
$ export AWS_DEFAULT_PROFILE=AccountAadmin
```

Per configurare AWS Tools for Windows PowerShell

1. Scarica e configura la AWS Tools for Windows PowerShell. Per istruzioni, vai alla sezione Installazione del AWS Tools for Windows PowerShell nella Guida per l'AWS Tools for Windows PowerShell utente.

   > **Note**
   >
   > Per caricare il AWS Tools for Windows PowerShell modulo, è necessario abilitare l'esecuzione PowerShell dello script. Per ulteriori informazioni, consulta Enable Script Execution nella Guida AWS Tools for Windows PowerShell per l'utente.

2. Per queste procedure dettagliate, è possibile specificare AWS le credenziali per sessione utilizzando il comando. `Set-AWSCredentials` Il comando salva le credenziali in uno store permanente (parametro `-StoreAs`).

   ```
   Set-AWSCredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -
   storeas string
   ```

3. Verificare la configurazione.

   - Per recuperare un elenco di comandi disponibili che puoi utilizzare per le operazioni di Amazon S3, `Get-Command` esegui il comando.

     ```
     Get-Command -module awspowershell -noun s3* -StoredCredentials string
     ```

   - Per recuperare un elenco di oggetti in un bucket, esegui il comando. `Get-S3Object`

     ```
     Get-S3Object -BucketName bucketname -StoredCredentials string
     ```

Per un elenco di comandi, vedere AWS Tools for PowerShell Cmdlet Reference.

Ora sei pronto per provare le procedure dettagliate. Segui i link forniti all'inizio di ogni sezione.

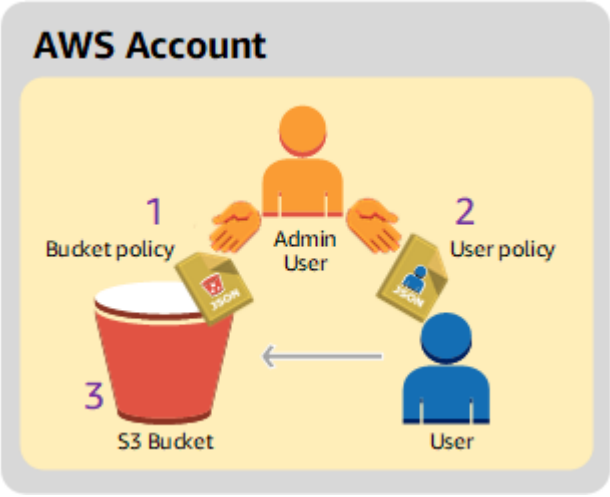
Esempio 1: il proprietario del bucket concede agli utenti le autorizzazioni per il bucket

> **Important**
>
> La concessione delle autorizzazioni ai ruoli IAM è una pratica migliore rispetto alla concessione delle autorizzazioni a singoli utenti. Per ulteriori informazioni su come concedere le autorizzazioni ai ruoli IAM, consulta. Comprensione delle autorizzazioni tra account e utilizzo dei ruoli IAM

Argomenti

In questa procedura dettagliata, un utente Account AWS possiede un bucket e l'account include un utente IAM. Per impostazione predefinita, l'utente non dispone di autorizzazioni. Per eseguire qualsiasi attività, l'account padre deve concedere le autorizzazioni all'utente. Il proprietario del bucket e l'account padre sono uguali. Pertanto, per concedere all'utente le autorizzazioni sul bucket, Account AWS possono utilizzare una policy del bucket, una policy utente o entrambe. Il proprietario dell'account concederà alcune autorizzazioni con una policy del bucket e altre con una policy utente.

La seguenti fasi riepilogano la procedura guidata:



1. L'amministratore dell'account crea una policy bucket per concedere un set di autorizzazioni all'utente.

2. L'amministratore dell'account collega una policy utente all'utente per concedere ulteriori autorizzazioni.
3. L'utente prova quindi le autorizzazioni concesse tramite la policy bucket e la policy utente.

Per questo esempio, avrai bisogno di un. Account AWS Anziché utilizzare le credenziali dell'utente root dell'account, sarà necessario creare un utente amministratore (consultare Informazioni sull'uso di un utente amministratore per creare risorse e concedere autorizzazioni). Ci riferiamo all'utente Account AWS e all'utente amministratore come illustrato nella tabella seguente.

ID account	Account denominato	Utente amministratore nell'account
1111-1111-1111	Account A	AccountAadmin

> **Note**
>
> L'utente amministratore in questo esempio è AccountAadmin, che si riferisce all'account A e non AccountAdmin.

Tutte le attività di creazione degli utenti e assegnazione delle autorizzazioni vengono effettuate nella AWS Management Console. Per verificare le autorizzazioni, la procedura dettagliata utilizza gli strumenti della riga di comando, AWS Command Line Interface (AWS CLI) e AWS Tools for Windows PowerShell quindi non è necessario scrivere alcun codice.

Preparazione della spiegazione passo per passo

1. Assicurati di avere un account Account AWS e che abbia un utente con privilegi di amministratore.

   a. Registrati per un Account AWS, se necessario. Si fa riferimento a questo account come Account A.

      i. Vai su https://aws.amazon.com/s3 e scegli Crea un AWS account.
      ii. Seguire le istruzioni su schermo.

      AWS ti avviserà via e-mail quando il tuo account sarà attivo e disponibile per l'uso.

b. Nell'Account A, crea un utente amministratore**`AccountAadmin`**. Utilizzando le credenziali dell'Account A, accedere alla console IAM ed effettuare quanto segue:

   i. Crea utente **`AccountAadmin`** e annota le credenziali di sicurezza dell'utente.

      Per istruzioni, consulta Creating an IAM user in your Account AWS nella IAM User Guide.

   ii. Concedi i privilegi di amministratore AccountAadminallegando una policy utente che dia accesso completo.

      Per le istruzioni, consulta Gestione di policy IAM nella Guida per l'utente di IAM.

   iii. Nota l'URL di accesso dell'utente IAM per. AccountAadmin che dovrà essere utilizzato per accedere alla AWS Management Console. Per ulteriori informazioni su dove trovare l'URL di accesso, consulta Accedere AWS Management Console come utente IAM nella IAM User Guide. Annota l'URL di ogni account.

2. Configura il AWS CLI o il AWS Tools for Windows PowerShell. Assicurati di salvare le credenziali dell'utente amministratore come segue:

   - Se usi il AWS CLI, crea un profilo nel file di configurazione. `AccountAadmin`
   - Se utilizzi il AWS Tools for Windows PowerShell, assicurati di memorizzare le credenziali per la sessione come. `AccountAadmin`

   Per istruzioni, consulta Configurazione degli strumenti per le procedure dettagliate.

Passaggio 1: Creare risorse nell'Account A e concedere le autorizzazioni

Utilizzando le credenziali dell'utente `AccountAadmin` nell'Account A e lo speciale URL di accesso utente IAM, accedi a AWS Management Console e procedi come segue:

1. Crea le risorse di un bucket e di un utente IAM

   a. Nella console di Amazon S3 creare un bucket. Nota Regione AWS in che modo hai creato il bucket. Per istruzioni, consulta Creazione di un bucket.

   b. Nella console IAM, procedi come segue:

      i. Crea un utente chiamato Dave.

Per step-by-step istruzioni, consulta Creazione di utenti IAM (console) nella Guida per l'utente IAM.

ii. Prendi nota delle `UserDave` credenziali.

iii. Prendi nota dell'Amazon Resource Name (ARN) per l'utente Dave. Nella console IAM, seleziona l'utente e la scheda Riepilogo fornisce l'ARN dell'utente.

2. Concedi le autorizzazioni.

Poiché il proprietario del bucket e l'account principale a cui appartiene l'utente sono gli stessi, Account AWS possono concedere le autorizzazioni all'utente utilizzando una policy del bucket, una politica utente o entrambe. come in questo esempio. Se l'oggetto è anche di proprietà dello stesso account, il proprietario del bucket può concedere le autorizzazioni per l'oggetto nella policy bucket (o una policy IAM).

a. Nella console di Amazon S3, collegare la seguente policy bucket a *`awsexamplebucket1`*.

La policy include due dichiarazioni.

- La prima istruzione concede a Dave le autorizzazioni per le operazioni sul bucket `s3:GetBucketLocation` e `s3:ListBucket`.
- La seconda istruzione concede l'autorizzazione `s3:GetObject`. Poiché l'Account A è anche proprietario dell'oggetto, l'amministratore dell'account può concedere l'autorizzazione `s3:GetObject`.

Nell'istruzione `Principal` Dave è identificato dall'ARN utente. Per ulteriori informazioni sugli elementi delle policy, consultare Politiche e autorizzazioni in 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. Creare una policy inline per l'utente Dave mediante la policy che segue. La policy concede a Dave l'autorizzazione `s3:PutObject`. È necessario aggiornare la policy specificando il nome del bucket.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "PermissionForObjectOperations",
         "Effect": "Allow",
         "Action": [
            "s3:PutObject"
         ],
         "Resource": [
            "arn:aws:s3:::awsexamplebucket1/*"
         ]
      }
   ]
}
```

Per istruzioni, consulta Managing IAMPolicies nella IAM User Guide. Tenere presente che è necessario accedere alla console tramite le credenziali dell'Account A.

Fase 2: testare le autorizzazioni

Utilizzando le credenziali di Dave, verificare che le autorizzazioni funzionino correttamente. È possibile utilizzare una delle due procedure di seguito.

Verifica le autorizzazioni utilizzando il AWS CLI

1. Aggiorna il file di AWS CLI configurazione aggiungendo il seguente `UserDaveAccountA` profilo. Per ulteriori informazioni, consulta Configurazione degli strumenti per le procedure dettagliate.

```
[profile UserDaveAccountA]
aws_access_key_id = access-key
aws_secret_access_key = secret-access-key
region = us-east-1
```

2. Verificare che Dave possa eseguire le operazioni autorizzate nella policy utente. Caricate un oggetto di esempio utilizzando il AWS CLI `put-object` comando seguente.

   Il parametro `--body` nel comando identifica il file di origine da caricare. Ad esempio, se il file si trova nella radice dell'unità C: su una Windows macchina, si specifica`c:\HappyFace.jpg`. Il parametro `--key` fornisce il nome della chiave dell'oggetto.

```
aws s3api put-object --bucket awsexamplebucket1 --key HappyFace.jpg --
body HappyFace.jpg --profile UserDaveAccountA
```

   Eseguite il AWS CLI comando seguente per ottenere l'oggetto.

```
aws s3api get-object --bucket awsexamplebucket1 --key HappyFace.jpg OutputFile.jpg
 --profile UserDaveAccountA
```

Verifica le autorizzazioni utilizzando il AWS Tools for Windows PowerShell

1. Memorizza le credenziali di Dave come. `AccountADave` Queste credenziali vengono quindi utilizzate per PUT aggiungere un oggetto. `GET`

```
set-awscredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas
 AccountADave
```

2. Caricate un oggetto di esempio utilizzando il AWS Tools for Windows PowerShell `Write-S3Object` comando utilizzando le credenziali memorizzate dell'utente Dave.

```
Write-S3Object -bucketname awsexamplebucket1 -key HappyFace.jpg -file HappyFace.jpg
 -StoredCredentials AccountADave
```

Scaricare l'oggetto caricato in precedenza.

```
Read-S3Object -bucketname awsexamplebucket1 -key HappyFace.jpg -file Output.jpg -
StoredCredentials AccountADave
```

Esempio 2: il proprietario del bucket concede autorizzazioni per il bucket multiaccount

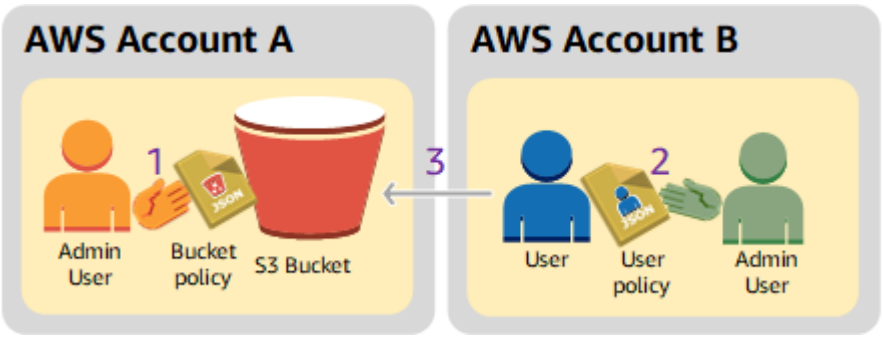
> **Important**
>
> Concedere le autorizzazioni ai ruoli IAM è una pratica migliore rispetto alla concessione delle autorizzazioni ai singoli utenti. Per informazioni su come effettuare questa operazione, consulta Comprensione delle autorizzazioni tra account e utilizzo dei ruoli IAM.

Argomenti

Un account A, ad Account AWS esempio, può concedere a un altro, l'account B Account AWS, l'autorizzazione ad accedere alle sue risorse come bucket e oggetti. L'Account B può quindi delegare queste autorizzazioni agli utenti nel proprio account. In questo scenario di esempio, il proprietario del bucket concede a un altro account le autorizzazioni multiaccount per eseguire specifiche operazioni nel bucket.

> **Note**
>
> L'account A può inoltre concedere le autorizzazioni a un utente dell'Account B mediante una policy di bucket. Tuttavia, l'utente avrà comunque bisogno dell'autorizzazione dell'account principale, l'account B, a cui appartiene l'utente, anche se l'account B non dispone delle autorizzazioni dell'account A. Finché l'utente dispone dell'autorizzazione sia del proprietario della risorsa che dell'account principale, l'utente sarà in grado di accedere alla risorsa.

Di seguito è riportato un riepilogo delle fasi della procedura guidata:



1. L'amministratore dell'Account A collega una policy di bucket che concede all'Account B autorizzazioni multiaccount per l'esecuzione di specifiche operazioni nel bucket.

   L'utente amministratore dell'Account B erediterà automaticamente le autorizzazioni.
2. L'utente amministratore dell'account B collega una policy utente all'utente per delegare le autorizzazioni ricevute dall'Account A.
3. L'utente dell'Account B fa quindi una verifica delle autorizzazioni accedendo a un oggetto nel bucket di proprietà dell'Account A.

Per questo utente, sono necessari due account. La tabella seguente mostra come viene fatto riferimento a questi account e ai relativi utenti amministratori. In conformità con le linee guida IAM (vediInformazioni sull'uso di un utente amministratore per creare risorse e concedere autorizzazioni), non utilizziamo le credenziali dell'utente root in questa procedura dettagliata. Viene invece creato un utente amministratore in ciascun account e le credenziali vengono utilizzate per la creazione di risorse e per concedere autorizzazioni a tali risorse.

Account AWS ID	Account denominato	Utente amministratore nell'account
1111-1111-1111	Account A	AccountAadmin
2222-2222-2222	Account B	AccountBadmin

Tutte le attività di creazione degli utenti e assegnazione delle autorizzazioni vengono effettuate nella AWS Management Console. Per verificare le autorizzazioni, la procedura dettagliata utilizza gli strumenti della riga di comando (AWS Command Line Interface CLI) e AWS Tools for Windows PowerShell quindi non è necessario scrivere alcun codice.

Preparazione della spiegazione passo per passo

1. Assicurati di averne due Account AWS e che ogni account abbia un utente amministratore, come mostrato nella tabella nella sezione precedente.

   a. Registrati per un Account AWS, se necessario.

   b. Utilizzando le credenziali dell'Account A, accedere alla console IAM per creare l'utente amministratore:

      i. Crea utente **`AccountAadmin`** e annota le credenziali di sicurezza. Per istruzioni, consulta Creazione di un utente IAM nell' Account AWS nella Guida per l'utente di IAM.

      ii. Concedi i privilegi di amministratore AccountAadminallegando una politica utente che dia accesso completo. Per istruzioni, consulta Gestione di policy IAM nella Guida per l'utente di IAM.

   c. Mentre sei nella console IAM, annota l'URL di accesso dell'utente IAM sulla dashboard. Tutti gli utenti dell'account devono utilizzare questo URL per accedere alla AWS Management Console.

      Per ulteriori informazioni, consulta In che modo gli utenti effettuano l'accesso al tuo account nella Guida per l'utente IAM.

   d. Ripeti il passaggio precedente utilizzando le credenziali dell'account B e crea l'utente amministratore. **`AccountBadmin`**

2. Imposta il AWS Command Line Interface (AWS CLI) o il. AWS Tools for Windows PowerShell Assicuratevi di salvare le credenziali dell'utente amministratore come segue:

   - Se utilizzate il AWS CLI, create due profili `AccountAadmin` e`AccountBadmin`, nel file di configurazione.
   - Se utilizzi il AWS Tools for Windows PowerShell, assicurati di memorizzare le credenziali per la sessione come e`AccountAadmin`. `AccountBadmin`

   Per istruzioni, consulta Configurazione degli strumenti per le procedure dettagliate.

3. Salvare le credenziali dell'utente amministratore, chiamate anche profili. È possibile utilizzare il nome del profilo anziché specificare le credenziali per ciascun comando immesso. Per ulteriori informazioni, consulta Configurazione degli strumenti per le procedure dettagliate.

   a. Aggiungi i profili nel file delle AWS CLI credenziali per ciascuno degli utenti amministratori `AccountAadmin` e `AccountBadmin` nei due account.

```
[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. Se utilizzi il AWS Tools for Windows PowerShell, esegui il comando seguente.

```
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
```

Fase 1: esecuzione delle attività per l'Account A

Passaggio 1.1: accedi a AWS Management Console

Utilizzando l'URL di accesso utente IAM per l'account A, accedi innanzitutto all'account AWS Management Console as AccountAadminuser. L'utente creerà un bucket e allegherà ad esso una policy.

Fase 1.2: creazione di un bucket

1. Nella console di Amazon S3 creare un bucket. Questo esercizio presuppone che il bucket sia stato creato negli Stati Uniti orientali (Virginia settentrionale) Regione AWS e abbia un nome. *DOC-EXAMPLE-BUCKET*

   Per istruzioni, consulta Creazione di un bucket.

2. Caricare un oggetto campione nel bucket.

   Per istruzioni, vai su Fase 2: Carica un oggetto nel tuo bucket.

Fase 1.3: collegare una policy del bucket per concedere autorizzazioni tra account all'Account B

La policy bucket concede le `s3:ListBucket` autorizzazioni `s3:GetLifecycleConfiguration` e all'account B. Si presume che tu abbia ancora effettuato l'accesso alla console utilizzando le credenziali utente. AccountAadmin

1. Collegare la seguente policy di bucket a *DOC-EXAMPLE-BUCKET*. La policy concede all'Account B autorizzazioni per le operazioni `s3:GetLifecycleConfiguration` e `s3:ListBucket`.

   Per istruzioni, consulta Aggiunta di una policy di bucket utilizzando la console di Amazon S3.

   ```
   {
      "Version": "2012-10-17",
      "Statement": [
         {
            "Sid": "Example permissions",
            "Effect": "Allow",
            "Principal": {
               "AWS": "arn:aws:iam::AccountB-ID:root"
            },
            "Action": [
               "s3:GetLifecycleConfiguration",
               "s3:ListBucket"
            ],
            "Resource": [
               "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
            ]
         }
      ]
   }
   ```

2. Verifica che l'account B (e quindi il relativo utente amministratore) sia in grado di eseguire le operazioni.

   - Verificare utilizzando il AWS CLI

     ```
     aws s3 ls s3://DOC-EXAMPLE-BUCKET --profile AccountBadmin
     aws s3api get-bucket-lifecycle-configuration --bucket DOC-EXAMPLE-BUCKET --
     profile AccountBadmin
     ```

   - Effettua la verifica utilizzando il AWS Tools for Windows PowerShell

```
get-s3object -BucketName DOC-EXAMPLE-BUCKET -StoredCredentials AccountBadmin
get-s3bucketlifecycleconfiguration -BucketName DOC-EXAMPLE-BUCKET -
StoredCredentials AccountBadmin
```

Fase 2: esecuzione delle attività per l'Account B

A questo punto l'amministratore dell'Account B crea un utente, Dave, al quale delega le autorizzazioni ricevute dall'Account A.

Passaggio 2.1: accedi a AWS Management Console

Utilizzando l'URL di accesso utente IAM per l'account B, accedi innanzitutto all'AccountBadminaccount AWS Management Console as.

Fase 2.2: creazione dell'utente Dave nell'Account B

Nella console IAM, crea un utente,**`Dave`**.

Per le istruzioni, consulta Creazione di utenti IAM (console) nella Guida per l'utente di IAM.

Fase 2.3: delega delle autorizzazioni all'utente Dave

Creare una policy inline per l'utente Dave mediante la policy che segue. Sarà necessario aggiornare la policy specificando il nome del bucket.

Si presume che tu abbia effettuato l'accesso alla console utilizzando le credenziali AccountBadminutente.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Example",
         "Effect": "Allow",
         "Action": [
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
         ]
      }
   ]
```

```
}
```

Per le istruzioni, consulta Gestione di policy IAM nella Guida per l'utente di IAM.

Fase 2.4: testare le autorizzazioni

Ora l'utente Dave dell'Account B può elencare il contenuto di *DOC-EXAMPLE-BUCKET* di proprietà dell'Account A. È possibile verificare le autorizzazioni mediante una delle procedure descritte di seguito.

Verifica le autorizzazioni utilizzando il AWS CLI

1. Aggiungi il `UserDave` profilo al file di AWS CLI configurazione. Per ulteriori informazioni sul file di configurazione, consulta Configurazione degli strumenti per le procedure dettagliate.

   ```
   [profile UserDave]
   aws_access_key_id = access-key
   aws_secret_access_key = secret-access-key
   region = us-east-1
   ```

2. Al prompt dei comandi, inserisci il seguente AWS CLI comando per verificare che Dave possa ora ottenere un elenco di oggetti dall'account di *DOC-EXAMPLE-BUCKET* proprietà dell'Account A. Nota che il comando specifica il profilo. `UserDave`

   ```
   aws s3 ls s3://DOC-EXAMPLE-BUCKET --profile UserDave
   ```

   Dave non ha altre autorizzazioni. Quindi, se prova qualsiasi altra operazione, ad esempio la seguente `get-bucket-lifecycle` configurazione, Amazon S3 restituisce l'autorizzazione negata.

   ```
   aws s3api get-bucket-lifecycle-configuration --bucket DOC-EXAMPLE-BUCKET --profile
    UserDave
   ```

Verifica le autorizzazioni utilizzando AWS Tools for Windows PowerShell

1. Memorizza le credenziali di Dave come. `AccountBDave`

   ```
   set-awscredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas
    AccountBDave
   ```

2. Provare a utilizzare il comando List Bucket.

```
get-s3object -BucketName DOC-EXAMPLE-BUCKET -StoredCredentials AccountBDave
```

Dave non ha altre autorizzazioni. Quindi, se prova qualsiasi altra operazione, ad esempio la seguente, `get-s3bucketlifecycleconfiguration` Amazon S3 restituisce l'autorizzazione negata.

```
get-s3bucketlifecycleconfiguration -BucketName DOC-EXAMPLE-BUCKET -
StoredCredentials AccountBDave
```

Fase 3: (facoltativo) provare un rifiuto esplicito

È possibile ottenere le autorizzazioni utilizzando una lista di controllo degli accessi (ACL), una policy bucket o una policy utente. Tuttavia, se esiste una negazione esplicita impostata da una policy bucket o da una policy utente, la negazione esplicita ha la precedenza su qualsiasi altra autorizzazione. Per il test, aggiorna la policy del bucket e nega esplicitamente l'autorizzazione all'Account B. `s3:ListBucket` La politica concede anche l'autorizzazione. `s3:ListBucket` Tuttavia, la negazione esplicita ha la precedenza e l'Account B o gli utenti dell'Account B non saranno in grado di elencare oggetti in esso. *DOC-EXAMPLE-BUCKET*

1. Utilizzando le credenziali dell'utente `AccountAadmin` nell'Account A, sostituisci la policy bucket con la seguente.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Example permissions",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountB-ID:root"
         },
         "Action": [
            "s3:GetLifecycleConfiguration",
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
```

```
            ]
        },
        {
            "Sid": "Deny permission",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:root"
            },
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
            ]
        }
    ]
}
```

2. Ora, se provi a creare una lista dei desideri utilizzando `AccountBadmin` le credenziali, l'accesso viene negato.

   - Usando AWS CLI, esegui il seguente comando:

     ```
     aws s3 ls s3://DOC-EXAMPLE-BUCKET --profile AccountBadmin
     ```

   - Utilizzando il AWS Tools for Windows PowerShell, esegui il comando seguente:

     ```
     get-s3object -BucketName DOC-EXAMPLE-BUCKET -StoredCredentials AccountBDave
     ```

Fase 4: pulizia

1. Dopo aver terminato il test, puoi effettuare le seguenti operazioni per ripulire:

   - Accedi a AWS Management Console (AWS Management Console) utilizzando le credenziali dell'Account A ed esegui le seguenti operazioni:

     - Nella console di Amazon S3 rimuovere la policy del bucket collegata a *DOC-EXAMPLE-BUCKET*. Nella sezione Properties (Proprietà) del bucket, eliminare la policy nella sezione Permissions (Autorizzazioni).
     - Se il bucket è stato creato per questo esercizio, nella console di Amazon S3 eliminare gli oggetti e quindi il bucket.

- Nella console IAM, rimuovi l'`AccountAadmin`utente.

2. Accedi alla console IAM utilizzando le credenziali dell'account B. Elimina utente`AccountBadmin`. Per step-by-step istruzioni, consulta Eliminazione di un utente IAM nella Guida per l'utente IAM.

Esempio 3: il proprietario del bucket concede autorizzazioni per gli oggetti che non sono di sua proprietà

> **Important**
>
> Concedere le autorizzazioni ai ruoli IAM è una pratica migliore rispetto alla concessione delle autorizzazioni ai singoli utenti. Per informazioni su come effettuare questa operazione, consulta Comprensione delle autorizzazioni tra account e utilizzo dei ruoli IAM.

Argomenti

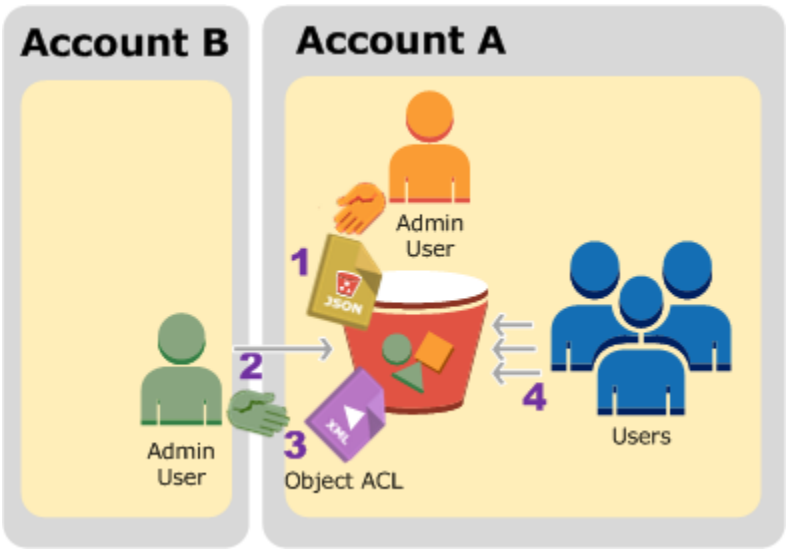
Lo scenario di questo esempio è che il proprietario del bucket desidera concedere l'autorizzazione per accedere agli oggetti, ma il proprietario del bucket non possiede tutti gli oggetti nel bucket. In questo esempio, il proprietario del bucket tenta di concedere autorizzazioni agli utenti nel proprio account.

Il proprietario di un bucket può consentire ad altri Account AWS di caricare oggetti. Per impostazione predefinita, il proprietario del bucket non possiede oggetti scritti su un bucket da un altro Account AWS. Gli oggetti sono di proprietà degli account che li scrivono in un bucket S3. Se il proprietario del bucket non possiede oggetti nel bucket, deve prima concedere l'autorizzazione al proprietario del bucket utilizzando una lista di controllo dell'accesso agli oggetti (ACL). Quindi, il proprietario del bucket può concedere le autorizzazioni a un oggetto di cui non è proprietario. Per ulteriori informazioni, consulta Proprietà di bucket e oggetti di Amazon S3.

Se il proprietario del bucket esegue l'impostazione proprietario del bucket applicato per S3 Object Ownership per il bucket, il proprietario del bucket possiederà tutti gli oggetti nel bucket, inclusi gli oggetti scritti da un altro Account AWS. Questo approccio risolve il problema che gli oggetti non sono di proprietà del proprietario del bucket. Quindi, puoi delegare le autorizzazioni agli utenti nel tuo account o ad altri Account AWS.

> **Note**
>
> S3 Proprietà dell'oggetto è un'impostazione a livello di bucket Amazon S3 che è possibile utilizzare per controllare la proprietà degli oggetti caricati nel bucket e per disabilitare o abilitare le liste di controllo degli accessi (ACL). Per impostazione predefinita, Proprietà dell'oggetto è impostata su Proprietario del bucket applicato e tutte le liste di controllo degli accessi (ACL) sono disabilitate. Quando le ACL sono disabilitate, il proprietario del bucket dispone di tutti gli oggetti nel bucket e gestisce l'accesso ad essi in maniera esclusiva utilizzando policy di gestione dell'accesso.
> La maggior parte degli attuali casi d'uso in Amazon S3 non richiede più l'uso delle ACL. È consigliabile mantenere le ACL disabilitate, tranne nelle circostanze in cui è necessario controllare individualmente l'accesso per ciascun oggetto. Con le ACL disabilitate, puoi utilizzare le policy per controllare l'accesso a tutti gli oggetti nel bucket, a prescindere da chi ha caricato gli oggetti nel bucket. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

In questo esempio supponiamo che il proprietario del bucket non abbia applicato l'impostazione proprietario del bucket applicato per Object Ownership. Il proprietario del bucket delega queste autorizzazioni agli utenti nel suo account. Di seguito è riportato un riepilogo dei passaggi dettagliati:



1. L'utente amministratore dell'Account A collega una policy di bucket con due istruzioni.
   - Concedere all'Account B autorizzazioni multiaccount per caricare oggetti.
   - Consentire a un utente nel proprio account di accedere agli oggetti nel bucket.
2. L'utente amministratore dell'account B carica gli oggetti nel bucket di proprietà dell'Account A.
3. L'amministratore dell'Account B aggiorna l'ACL dell'oggetto e concede al proprietario del bucket l'autorizzazione al controllo completo sull'oggetto.
4. L'utente dell'Account A fa una verifica accedendo agli oggetti nel bucket, indipendentemente da chi ne ha la proprietà.

Per questo utente, sono necessari due account. La tabella seguente mostra come viene fatto riferimento a questi account e agli utenti amministratori degli account: In questa spiegazione passo per passo, non si utilizzano le credenziali dell'utente root dell'account, in base a quanto riportato nelle linee guida IAM consigliate. Per ulteriori informazioni, consulta Informazioni sull'uso di un utente amministratore per creare risorse e concedere autorizzazioni. Viene invece creato un utente amministratore in ciascun account e le credenziali vengono utilizzate per la creazione di risorse e per concedere autorizzazioni a tali risorse

Account AWS ID	Account denominato	Amministratore nell'account
1111-1111-1111	Account A	AccountAadmin
2222-2222-2222	Account B	AccountBadmin

Tutte le attività di creazione degli utenti e assegnazione delle autorizzazioni vengono effettuate nella AWS Management Console. Per verificare le autorizzazioni, la procedura dettagliata utilizza gli strumenti della riga di comando, AWS Command Line Interface (AWS CLI) e AWS Tools for Windows PowerShell quindi non è necessario scrivere alcun codice.

Fase 0: preparazione della procedura guidata

1. Assicurati di averne due Account AWS e che ogni account abbia un amministratore, come mostrato nella tabella nella sezione precedente.

   a. Iscriviti a un Account AWS, se necessario.

   b. Utilizzando le credenziali dell'Account A, accedi alla console IAM e procedi come segue per creare un utente amministratore:

      - Crea utente **`AccountAadmin`** e annota le credenziali di sicurezza dell'utente. Per ulteriori informazioni sull'aggiunta di utenti, consulta Creazione di un utente IAM nell' Account AWS nella Guida per l'utente di IAM.
      - Concedi le autorizzazioni di amministratore AccountAadminallegando una politica utente che dia accesso completo. Per le istruzioni, consulta Gestione di policy IAM nella Guida per l'utente di IAM.
      - Nella dashboard della console IAM, annota l'URL di accesso utente IAM. Gli utenti di questo account devono utilizzare questo URL per accedere alla AWS Management Console. Per ulteriori informazioni, consulta In che modo gli utenti effettuano l'accesso al tuo account nella Guida per l'utente IAM.

   c. Ripeti il passaggio precedente utilizzando le credenziali dell'account B e crea l'utente amministratore. **`AccountBadmin`**

2. Configura AWS CLI o gli strumenti per Windows. PowerShell Assicurati di salvare le credenziali di amministratore nel modo seguente:

   - Se usi il AWS CLI, crea due profili `AccountAadmin` e`AccountBadmin`, nel file di configurazione.
   - Se utilizzi gli Strumenti per Windows PowerShell, assicurati di memorizzare le credenziali per la sessione come `AccountAadmin` e. `AccountBadmin`

   Per istruzioni, consulta Configurazione degli strumenti per le procedure dettagliate.

Fase 1: esecuzione delle attività per l'Account A

Esegui le operazioni riportate di seguito per l'Account A:

Fase 1.1: Accesso alla console

Utilizzando l'URL di accesso utente IAM per l'account A, accedi all'utente AWS Management Console as**`AccountAadmin`**. L'utente creerà un bucket e allegherà ad esso una policy.

Fase 1.2: Creazione di un bucket e di un utente e aggiunta di una policy di bucket che concede le autorizzazioni utente

1. Nella console di Amazon S3 creare un bucket. Questo esercizio presuppone che il bucket sia stato creato negli Stati Uniti orientali (Virginia settentrionale) Regione AWS e che il nome sia. *`example-s3-bucket1`*

   Per istruzioni, consulta Creazione di un bucket.
2. Nella console IAM, crea un utente. **`Dave`**

   Per step-by-step istruzioni, consulta Creazione di utenti IAM (console) nella Guida per l'utente IAM.
3. Prendi nota delle credenziali dell'utente Dave.
4. Nella console di Amazon S3 collegare la seguente policy del bucket a *`example-s3-bucket1`*. Per istruzioni, consulta Aggiunta di una policy di bucket utilizzando la console di Amazon S3. Seguire le fasi per l'aggiunta di una policy di bucket. Per informazioni su come trovare gli ID degli account, vedi Ricerca del tuo Account AWS ID.

   La policy concede all'Account B le autorizzazioni `s3:PutObject` e `s3:ListBucket`. La politica concede inoltre l'`s3:GetObject`autorizzazione `Dave` all'utente.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Statement1",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountB-ID:root"
         },
         "Action": [
```

```
            "s3:PutObject",
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::example-s3-bucket1/*",
            "arn:aws:s3:::example-s3-bucket1"
         ]
      },
      {
         "Sid": "Statement3",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
         },
         "Action": [
            "s3:GetObject"
         ],
         "Resource": [
            "arn:aws:s3:::example-s3-bucket1/*"
         ]
      }
   ]
}
```

Fase 2: esecuzione delle attività per l'Account B

Ora che l'Account B dispone delle autorizzazioni per eseguire operazioni sul bucket dell'Account A, l'amministratore dell'Account B esegue le seguenti operazioni:

- Carica un oggetto nel bucket dell'Account A
- Aggiunge una concessione nell'ACL dell'oggetto per consentire all'account A, il proprietario del bucket, il pieno controllo

Utilizzando il AWS CLI

1. Utilizzando il `put-object` AWS CLI comando, caricate un oggetto. Il parametro `--body` nel comando identifica il file di origine da caricare. Ad esempio, se il file si trova sull'`C:`unità di una Windows macchina, specificate`c:\HappyFace.jpg`. Il parametro `--key` fornisce il nome della chiave dell'oggetto.

```
aws s3api put-object --bucket example-s3-bucket1 --key HappyFace.jpg --body
 HappyFace.jpg --profile AccountBadmin
```

2. Aggiungere un'autorizzazione nell'ACL dell'oggetto per concedere controllo completo dell'oggetto al proprietario del bucket. Per informazioni su come trovare un ID utente canonico, consulta Trova l'ID utente canonico per te Account AWS nella AWS Account Management Reference Guide.

```
aws s3api put-object-acl --bucket example-s3-bucket1 --key HappyFace.jpg --grant-
full-control id="AccountA-CanonicalUserID" --profile AccountBadmin
```

Utilizzo degli strumenti per Windows PowerShell

1. Utilizzando il `Write-S3Object` comando, caricate un oggetto.

```
Write-S3Object -BucketName example-s3-bucket1 -key HappyFace.jpg -file
 HappyFace.jpg -StoredCredentials AccountBadmin
```

2. Aggiungere un'autorizzazione nell'ACL dell'oggetto per concedere controllo completo dell'oggetto al proprietario del bucket.

```
Set-S3ACL -BucketName example-s3-bucket1 -Key HappyFace.jpg -CannedACLName "bucket-
owner-full-control" -StoredCreden
```

Fase 3: testare le autorizzazioni

A questo punto, verifica se l'utente Dave nell'Account A ha accesso all'oggetto di proprietà dell'Account B.

Usando il AWS CLI

1. Aggiungi le credenziali dell'utente Dave al file di AWS CLI configurazione e crea un nuovo profilo,. `UserDaveAccountA` Per ulteriori informazioni, consulta Configurazione degli strumenti per le procedure dettagliate.

```
[profile UserDaveAccountA]
aws_access_key_id = access-key
aws_secret_access_key = secret-access-key
```

```
region = us-east-1
```

2. Esegui il comando della CLI di `get-object` per scaricare `HappyFace.jpg` e salvarlo in locale. Le credenziali dell'utente Dave vengono fornite aggiungendo il parametro `--profile`.

```
aws s3api get-object --bucket example-s3-bucket1 --key HappyFace.jpg Outputfile.jpg
 --profile UserDaveAccountA
```

Utilizzo degli strumenti per Windows PowerShell

1. Archivia AWS le credenziali dell'utente Dave, come`UserDaveAccountA`, nell'archivio persistente.

```
Set-AWSCredentials -AccessKey UserDave-AccessKey -SecretKey UserDave-
SecretAccessKey -storeas UserDaveAccountA
```

2. Esegui il comando `Read-S3Object` per scaricare l'oggetto `HappyFace.jpg` e salvarlo in locale. Le credenziali dell'utente Dave vengono fornite aggiungendo il parametro `-StoredCredentials`.

```
Read-S3Object -BucketName example-s3-bucket1 -Key HappyFace.jpg -file HappyFace.jpg
  -StoredCredentials UserDaveAccountA
```

Fase 4: pulizia

1. Dopo aver terminato il test, puoi effettuare le seguenti operazioni per ripulire:

   - Accedi alla AWS Management Console utilizzando le credenziali dell'Account A e procedere come di seguito:

     - *Nella console Amazon S3, rimuovi la policy bucket allegata a example-s3-bucket1.* Nella sezione Properties (Proprietà) del bucket, eliminare la policy nella sezione Permissions (Autorizzazioni).
     - Se il bucket è stato creato per questo esercizio, nella console di Amazon S3 eliminare gli oggetti e quindi il bucket.
     - Nella console IAM, rimuovi l'utente. AccountAadmin Per step-by-step istruzioni, consulta Eliminazione di un utente IAM nella Guida per l'utente IAM.

2. Accedere alla AWS Management Console tramite le credenziali dell'Account B. Nella console IAM, elimina l'utente AccountBadmin.

Esempio 4: il proprietario del bucket concede l'autorizzazione di più account a oggetti di cui non è proprietario

Argomenti

In questo scenario di esempio, possiedi un bucket e hai consentito ad altri di caricare oggetti. Account AWS Se è stata applicata l'impostazione proprietario del bucket applicato per S3 Object Ownership per il bucket, si avrà proprietà di tutti gli oggetti nel bucket, inclusi gli oggetti scritti da un altro Account AWS. Questo approccio risolve il problema che gli oggetti non sono di proprietà dell'utente, il proprietario del bucket. Quindi, puoi delegare le autorizzazioni agli utenti nel tuo account o ad altri Account AWS. Supponiamo che l'impostazione proprietario del bucket applicato per S3 Object Ownership non sia abilitata. In altre parole, il bucket può avere oggetti di proprietà di altri Account AWS .

Ora, si supponga che, in qualità di proprietario del bucket, si debbano concedere autorizzazioni multiaccount per gli oggetti a un utente di un altro account, indipendentemente dall'utente a cui appartengono. Ad esempio, l'utente potrebbe essere un'applicazione per la fatturazione che deve accedere ai metadata dell'oggetto. Esistono due problemi principali:

- Il proprietario del bucket non dispone delle autorizzazioni per gli oggetti creati dagli altri Account AWS. Affinché il proprietario del bucket conceda le autorizzazioni su oggetti che non possiede, deve prima concedere l'autorizzazione al proprietario del bucket. Il proprietario dell'oggetto è colui Account AWS che ha creato gli oggetti. Il proprietario del bucket può quindi delegare tali autorizzazioni.

- L'account proprietario del bucket può delegare le autorizzazioni agli utenti nel proprio account (vedi). Esempio 3: il proprietario del bucket concede autorizzazioni per gli oggetti che non sono di sua proprietà Tuttavia, l'account proprietario del bucket non può delegare le autorizzazioni ad altri Account AWS perché la delega tra account non è supportata.

In questo scenario, il proprietario del bucket può creare un ruolo AWS Identity and Access Management (IAM) con l'autorizzazione ad accedere agli oggetti. Quindi, il proprietario del bucket può concedere un'altra Account AWS autorizzazione per assumere il ruolo, consentendogli temporaneamente di accedere agli oggetti nel bucket.

> **Note**
>
> S3 Proprietà dell'oggetto è un'impostazione a livello di bucket Amazon S3 che è possibile utilizzare per controllare la proprietà degli oggetti caricati nel bucket e per disabilitare o abilitare le liste di controllo degli accessi (ACL). Per impostazione predefinita, Proprietà dell'oggetto è impostata su Proprietario del bucket applicato e tutte le liste di controllo degli accessi (ACL) sono disabilitate. Quando le ACL sono disabilitate, il proprietario del bucket dispone di tutti gli oggetti nel bucket e gestisce l'accesso ad essi in maniera esclusiva utilizzando policy di gestione dell'accesso.
> La maggior parte degli attuali casi d'uso in Amazon S3 non richiede più l'uso delle ACL. È consigliabile mantenere le ACL disabilitate, tranne nelle circostanze in cui è necessario controllare individualmente l'accesso per ciascun oggetto. Con le ACL disabilitate, puoi utilizzare le policy per controllare l'accesso a tutti gli oggetti nel bucket, a prescindere da chi ha caricato gli oggetti nel bucket. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

Comprensione delle autorizzazioni tra account e utilizzo dei ruoli IAM

I ruoli IAM consentono diversi scenari per la delega dell'accesso alle risorse. Uno degli scenari principali è l'accesso multiaccount. In questo esempio, il proprietario del bucket, l'Account A, utilizza un ruolo IAM per delegare temporaneamente l'accesso agli oggetti tra più account agli utenti di un altro account Account AWS, l'Account C. A ogni ruolo IAM creato sono associate le seguenti due policy:

- Una policy di fiducia che ne identifica un'altra Account AWS che può assumere il ruolo.

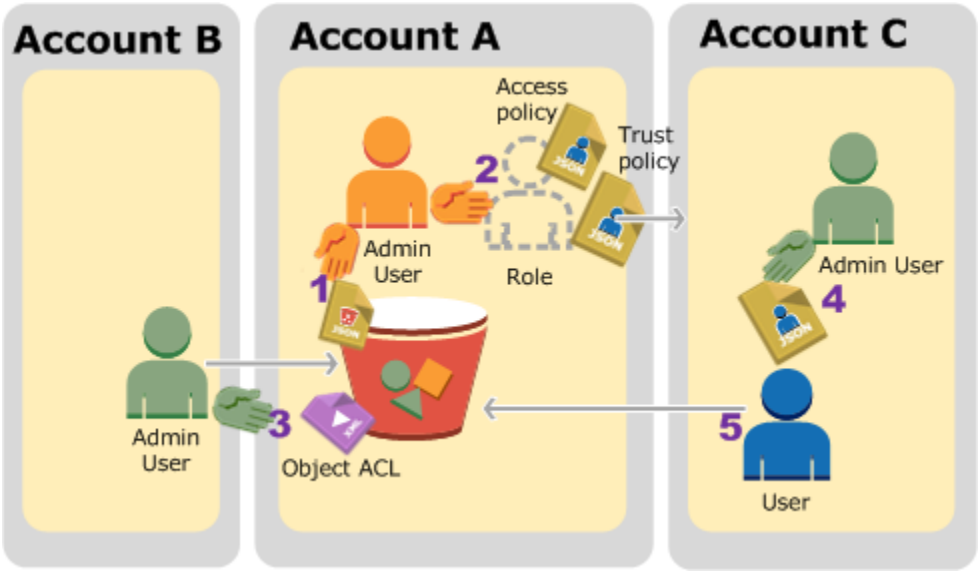
- Una policy di accesso per la definizione delle autorizzazioni consentite quando qualcuno assume il ruolo, ad esempio `s3:GetObject`. Per un elenco delle autorizzazioni che è possibile specificare in una policy, consulta Azioni politiche per Amazon S3.

La Account AWS persona identificata nella politica di fiducia concede quindi all'utente l'autorizzazione ad assumere il ruolo. L'utente può quindi accedere agli oggetti nel modo seguente:

- Assumere il ruolo e, in risposta, ottenere le credenziali di sicurezza temporanee.
- Accedere agli oggetti nel bucket utilizzando le credenziali di sicurezza temporanee.

Per ulteriori informazioni sui ruoli IAM, consulta Ruoli IAM nella Guida per l'utente di IAM.

Di seguito è riportato un riepilogo dei passaggi dettagliati:



1. L'utente amministratore dell'account A collega una policy di bucket che concede all'Account B un'autorizzazione condizionale per caricare gli oggetti.
2. L'amministratore dell'Account A crea un ruolo IAM per stabilire l'attendibilità con l'Account C, pertanto gli utenti in quell'account possono accedere all'Account A. La policy di accesso collegata al ruolo limita le operazioni consentite all'utente nell'Account C quando accede all'Account A.
3. L'amministratore dell'Account B carica un oggetto nel bucket di proprietà dell'Account A, concedendo al proprietario del bucket un'autorizzazione al controllo completo.
4. L'amministratore dell'account C crea un utente e collega una policy utente che gli consente di assumere il ruolo.

5. L'utente nell'Account C per prima cosa assume il ruolo, che restituisce all'utente credenziali di sicurezza temporanee. Mediante tali credenziali temporanee, l'utente accede quindi agli oggetti nel bucket.

Per questo esempio sono necessari tre account. La tabella seguente mostra come viene fatto riferimento a questi account e agli utenti amministratori degli account: In conformità con le linee guida IAM (vediInformazioni sull'uso di un utente amministratore per creare risorse e concedere autorizzazioni), non utilizziamo le Utente root dell'account AWS credenziali in questa procedura dettagliata. Viene invece creato un utente amministratore in ciascun account e le credenziali vengono utilizzate per la creazione di risorse e per concedere autorizzazioni a tali risorse.

Account AWS ID	Account denominato	Utente amministratore nell'account
1111-1111-1111	Account A	AccountAadmin
2222-2222-2222	Account B	AccountBadmin
3333-3333-3333	Account C	AccountCadmin

Fase 0: preparazione della procedura guidata

> **Note**
>
> Potresti voler aprire un editor di testo e annotare alcune informazioni man mano che procedi. In particolare, è necessario disporre di ID account, ID utenti canonici, URL di accesso utente IAM per la connessione di ciascun account alla console, Amazon Resource Names (ARN) degli utenti IAM e ruoli.

1. Assicurati di averne tre Account AWS e che ogni account abbia un utente amministratore, come mostrato nella tabella nella sezione precedente.

   a. Iscriviti a Account AWS, se necessario. Si fa riferimento a questi account come Account A, Account B e Account C.

b. Utilizzando le credenziali dell'Account A, accedere alla console IAM ed effettuare quanto segue per creare un utente amministratore:

   - Crea utente **`AccountAadmin`** e annota le sue credenziali di sicurezza. Per ulteriori informazioni sull'aggiunta di utenti, consulta Creazione di un utente IAM nell' Account AWS nella Guida per l'utente di IAM.
   - Concedi i privilegi di amministratore AccountAadminallegando una politica utente che dia accesso completo. Per le istruzioni, consulta Gestione di policy IAM nella Guida per l'utente di IAM.
   - Nella dashboard della console IAM, annota l'URL di accesso utente IAM. Gli utenti di questo account devono utilizzare questo URL per accedere alla AWS Management Console. Per ulteriori informazioni, consulta Accedere AWS Management Console come utente IAM nella Guida per l'utente IAM.

c. Ripetere la fase precedente per creare utenti amministratore nell'Account B e nell'Account C.

2. Per l'Account C, annota l'ID utente canonico.

   Quando si crea un ruolo IAM nell'Account A, la policy di attendibilità concede all'Account C l'autorizzazione per assumere il ruolo mediante la specifica dell'ID account. È possibile trovare informazioni sull'account come indicato di seguito:

   a. Usa il tuo Account AWS ID o alias dell'account, il tuo nome utente IAM e la password per accedere alla console Amazon S3.

   b. Scegliere il nome di un bucket Amazon S3 per visualizzare i relativi dettagli.

   c. Selezionare la scheda Permissions (Autorizzazioni) e selezionare Access Control List (Lista di controllo accessi).

   d. Nella sezione Accesso per il tuo Account AWS, nella colonna Account è presente un identificatore lungo, come `c1daexampleaaf850ea79cf0430f33d72579fd1611c97f7ded193374c0b163b6`. Questo è il tuo ID utente canonico.

3. Quando si crea una policy di bucket, è necessario disporre delle informazioni seguenti. Prendi nota di questi valori:

   - ID utente canonico dell'Account A – Quando l'amministratore dell'Account A concede all'amministratore dell'Account B l'autorizzazione condizionale per il caricamento degli oggetti,

la condizione specifica l'ID utente canonico dell'utente dell'Account A che deve ottenere controllo completo degli oggetti.

> **Note**
>
> L'ID utente canonico è un concetto esclusivo di Amazon S3. Si tratta di una versione offuscata dell'ID account, composta da 64 caratteri.

- ARN utente per amministratore dell'account B: puoi trovare l'ARN dell'utente nella console IAM. Devi selezionare l'utente e trovare l'ARN dell'utente nella scheda Riepilogo.

  Nella policy bucket, concedi `AccountBadmin` l'autorizzazione a caricare oggetti e specifichi l'utente utilizzando l'ARN. Ecco un esempio di valore ARN:

  ```
  arn:aws:iam::AccountB-ID:user/AccountBadmin
  ```

4. Configura la AWS Command Line Interface (CLI) o la. AWS Tools for Windows PowerShell Assicurati di salvare le credenziali utente dell'amministratore come segue:

   - Se utilizzate il AWS CLI, create profili `AccountAadmin` e`AccountBadmin`, nel file di configurazione.
   - Se utilizzi il AWS Tools for Windows PowerShell, assicurati di memorizzare le credenziali per la sessione come e`AccountAadmin`. `AccountBadmin`

   Per istruzioni, consulta Configurazione degli strumenti per le procedure dettagliate.

Fase 1: eseguire le attività per l'Account A

In questo esempio, l'Account A è il proprietario del bucket. Quindi l'utente dell'Account AccountAadmin A eseguirà le seguenti operazioni:

- Creare un bucket.
- Allega una policy bucket che conceda all'amministratore dell'Account B l'autorizzazione a caricare oggetti.
- Crea un ruolo IAM che conceda all'Account C l'autorizzazione ad assumere il ruolo in modo che possa accedere agli oggetti nel bucket.

Passaggio 1.1: accedi a AWS Management Console

Utilizzando l'URL di accesso dell'utente IAM per l'account A, accedi prima all'account AWS Management Console come **`AccountAadmin`** utente. L'utente creerà un bucket e allegherà ad esso una policy.

Fase 1.2: creare un bucket e allegarlo alla policy di bucket

Nella console di Amazon S3 effettuare quanto segue:

1. Creare un bucket. In questo esercizio si presume che il nome del bucket sia *`example-s3-bucket1`*.

   Per istruzioni, consulta Creazione di un bucket.

2. Allega la seguente policy sui bucket. La politica concede l'autorizzazione condizionata all'amministratore dell'Account B per caricare oggetti.

   Aggiorna la politica fornendo i tuoi valori per *`example-s3-bucket1AccountB-ID`*, e il. *`CanonicalUserId-of-AWSaccountA-BucketOwner`*

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "111",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::example-s3-bucket1/*"
        },
        {
            "Sid": "112",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::example-s3-bucket1/*",
            "Condition": {
                "StringNotEquals": {
```

```
                    "s3:x-amz-grant-full-control": "id=CanonicalUserId-of-
AWSaccountA-BucketOwner"
                }
            }
        }
    ]
}
```

Fase 1.3: Creare un ruolo IAM per consentire l'accesso da più account all'Account C all'Account A

Nella console IAM, crea un ruolo IAM (**examplerole**) che conceda all'Account C l'autorizzazione ad assumere il ruolo. Assicurati di aver ancora effettuato l'accesso come amministratore dell'account A perché il ruolo deve essere creato nell'account A.

1. Prima di creare il ruolo, preparare la policy gestita che definisce le autorizzazioni richieste dal ruolo. La policy verrà collegata al ruolo in una fase successiva.

   a. Nel riquadro di navigazione a sinistra, scegli Politiche, quindi scegli Crea politica.

   b. Accanto a Create Your Own Policy (Crea la tua policy) scegli Select (Seleziona).

   c. Immettere **access-accountA-bucket** nel campo Policy Name (Nome policy).

   d. Copiare la seguente policy di accesso e incollarla nel campo Policy Document (Documento policy). La politica di accesso concede l'`s3:GetObject`autorizzazione al ruolo, quindi, quando l'utente dell'Account C assume il ruolo, può solo eseguire l'`s3:GetObject`operazione.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::example-s3-bucket1/*"
       }
     ]
   }
   ```

   e. Scegliere Create Policy (Crea policy).

   La nuova policy viene inserita nell'elenco delle policy gestite.

2. Nel riquadro di navigazione a sinistra, scegli Ruoli, quindi scegli Crea nuovo ruolo.
3. In Seleziona tipo di ruolo, seleziona Ruolo per l'accesso su più account, quindi scegli il pulsante Seleziona accanto a Fornisci l'accesso tra i Account AWS tuoi utenti.
4. Immettere l'ID account dell'Account C.

   Per questa procedura dettagliata, non è necessario richiedere agli utenti di disporre dell'autenticazione a più fattori (MFA) per assumere il ruolo, quindi lascia questa opzione deselezionata.
5. Scegli Passaggio successivo per impostare le autorizzazioni che verranno associate al ruolo.
6. Seleziona la casella di controllo accanto alla policy Access-AccountA-Bucket che hai creato, quindi scegli Passaggio successivo.

   Viene visualizzata la pagina Review (Revisione) che consente di confermare le impostazioni per il ruolo prima che venga creato. Questa pagina contiene una voce molto importante, ossia il collegamento che è possibile inviare agli utenti che hanno necessità di utilizzare questo ruolo. Gli utenti che utilizzano il link accedono direttamente alla pagina Cambia ruolo con i campi ID account e Nome ruolo già compilati. Puoi anche vedere questo link più avanti nella pagina di riepilogo del ruolo per qualsiasi ruolo tra account.
7. Immetti `examplerole` il nome del ruolo, quindi scegli Passaggio successivo.
8. Dopo aver esaminato il ruolo, scegli Crea ruolo.

   Il ruolo `examplerole` viene visualizzato nell'elenco dei ruoli.
9. Scegli il nome del ruolo`examplerole`.
10. Selezionare la scheda Trust Relationships (Relazioni di trust).
11. Scegli Mostra il documento della politica e verifica che la politica di fiducia mostrata corrisponda alla seguente politica.

    Le seguente policy di attendibilità stabilisce l'attendibilità con l'Account C, consentendogli di eseguire l'operazione `sts:AssumeRole`. Per ulteriori informazioni, consulta AssumeRole nella documentazione di riferimento dell'API AWS Security Token Service .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
```

```
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountC-ID:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

12. Prendi nota dell'Amazon Resource Name (ARN) del `examplerole` ruolo che hai creato.

    Nelle fasi successive verrà illustrato come allegare una policy utente per consentire all'utente IAM di assumere questo ruolo e come identificare il ruolo mediante il valore ARN.

Fase 2: esecuzione delle attività per l'Account B

Il bucket di esempio di proprietà dell'Account A necessita di oggetti di proprietà di altri account. In questa fase, l'amministratore dell'Account B carica un oggetto mediante gli strumenti a riga di comando.

- Utilizzando il `put-object` AWS CLI comando, carica un oggetto su. *`example-s3-bucket1`*

  ```
  aws s3api put-object --bucket example-s3-bucket1 --key HappyFace.jpg --
  body HappyFace.jpg --grant-full-control id="canonicalUserId-ofTheBucketOwner" --
  profile AccountBadmin
  ```

  Tieni presente quanto segue:

  - Il `--Profile` parametro specifica il `AccountBadmin` profilo, quindi l'oggetto è di proprietà dell'account B.
  - Il parametro `grant-full-control` concede al proprietario del bucket l'autorizzazione al controllo completo sull'oggetto, come richiesto dalla policy di bucket.
  - Il parametro `--body` identifica il file di origine da caricare. Ad esempio, se il file si trova nell'unità C: di un Windows computer, si specifica`c:\HappyFace.jpg`.

Passaggio 3: Esegui le attività relative all'account C

Nei passaggi precedenti, l'Account A ha già creato un ruolo`examplerole`, stabilendo un rapporto di fiducia con l'Account C. Questo ruolo consente agli utenti dell'Account C di accedere

all'Account A. In questa fase, l'amministratore dell'Account C crea un utente (Dave) e gli delega l'`sts:AssumeRole`autorizzazione ricevuta dall'Account A. Questo approccio consente a Dave di assumere `examplerole` e accedere temporaneamente all'Account A. La politica di accesso che l'Account A ha associato al ruolo limita ciò che Dave può fare quando accede all'Account A: In particolare, inserisci oggetti*`example-s3-bucket1`*.

Passaggio 3.1: Creare un utente nell'account C e delegare l'autorizzazione ad assumere examplerole

1. Utilizzando l'URL di accesso utente IAM per l'Account C, accedi innanzitutto all'utente AWS Management Console as**`AccountCadmin`**.

2. Nella console IAM, crea un utente, Dave.

   Per step-by-step istruzioni, consulta Creating IAM users (AWS Management Console) nella IAM User Guide.

3. Nota le credenziali di Dave. Dave dovrà utilizzare queste credenziali per assumere il ruolo `examplerole`.

4. Crea una policy in linea per l'utente Dave IAM per delegare l'`sts:AssumeRole`autorizzazione a Dave sul ruolo nell'Account A. `examplerole`

   a. Nel riquadro di navigazione sinistro, scegli Utenti.

   b. Scegli il nome utente Dave.

   c. Nella pagina dei dettagli dell'utente, selezionare la scheda Permissions (Autorizzazioni), quindi espandere la sezione Inline Policies (Policy inline).

   d. Scegliere click here (fai clic qui) oppure Create User Policy (Crea policy di utente).

   e. Scegliere Custom Policy (Policy personalizzata) quindi Select (Seleziona).

   f. Immettere un nome per la policy nel campo Policy Name (Nome policy).

   g. Copiare la seguente policy nel campo Policy Document (Documento policy).

      È necessario aggiornare la politica fornendo il*`AccountA-ID`*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["sts:AssumeRole"],
      "Resource": "arn:aws:iam::AccountA-ID:role/examplerole"
```

```
    }
  ]
}
```

h. Scegli Apply Policy (Applica policy).

5. Salva le credenziali di Dave nel file di configurazione di AWS CLI aggiungendo un altro profilo,. `AccountCDave`

```
[profile AccountCDave]
aws_access_key_id = UserDaveAccessKeyID
aws_secret_access_key = UserDaveSecretAccessKey
region = us-west-2
```

Fase 3.2: Assumere role (examplerole) e accedere agli oggetti

Ora Dave può accedere agli oggetti nel bucket di proprietà dell'Account A nel modo seguente:

- Per prima cosa, Dave assume il ruolo `examplerole` utilizzando le sue credenziali personali. Questa operazione restituirà le credenziali temporanee.
- Dave utilizzerà le credenziali temporanee per accedere agli oggetti nel bucket dell'Account A.

1. Al prompt dei comandi, eseguite il AWS CLI `assume-role` comando seguente utilizzando il `AccountCDave` profilo.

   È necessario aggiornare il valore ARN nel comando fornendo *`AccountA-ID`* dove `examplerole` è definito.

```
aws sts assume-role --role-arn arn:aws:iam::AccountA-ID:role/examplerole --profile
 AccountCDave --role-session-name test
```

   In risposta, AWS Security Token Service (AWS STS) restituisce credenziali di sicurezza temporanee (ID della chiave di accesso, chiave di accesso segreta e un token di sessione).

2. Salva le credenziali di sicurezza temporanee nel file di AWS CLI configurazione sotto il profilo. `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. Al prompt dei comandi, esegui il AWS CLI comando seguente per accedere agli oggetti utilizzando le credenziali temporanee. Ad esempio, il comando specifica l'API head-object per recuperare i metadata dell'oggetto per l'oggetto `HappyFace.jpg`.

```
aws s3api get-object --bucket example-s3-bucket1 --key HappyFace.jpg SaveFileAs.jpg
 --profile TempCred
```

Dal momento che la policy di accesso collegata a `examplerole` consente l'esecuzione delle operazioni, Amazon S3 elabora la richiesta. È possibile provare ad eseguire un'altra operazione su qualsiasi altro oggetto nel bucket.

Se provi qualsiasi altra azione, ad esempio, `get-object-acl` ti verrà negata l'autorizzazione perché al ruolo non è consentita tale azione.

```
aws s3api get-object-acl --bucket example-s3-bucket1 --key HappyFace.jpg --profile
 TempCred
```

È stato utilizzato l'utente Dave per assumere il ruolo e accedere all'oggetto mediante le credenziali temporanee. L'accesso agli oggetti in *`example-s3-bucket1`* può essere effettuato anche da un'applicazione nell'Account C. L'applicazione può ottenere credenziali di sicurezza temporanee e l'Account C può delegare all'applicazione le autorizzazioni per assumere `examplerole`.

Fase 4: pulizia

1. Dopo aver terminato il test, puoi effettuare le seguenti operazioni per ripulire:

   - Accedi alla AWS Management Console utilizzando le credenziali dell'Account A e procedere come di seguito:

     - Nella console di Amazon S3 rimuovere la policy del bucket collegata a *`example-s3-bucket1`*. Nella sezione Properties (Proprietà) del bucket, eliminare la policy nella sezione Permissions (Autorizzazioni).
     - Se il bucket è stato creato per questo esercizio, nella console di Amazon S3 eliminare gli oggetti e quindi il bucket.

- Nella console IAM, rimuovi l'account `examplerole` che hai creato nell'account A. Per step-by-step istruzioni, consulta Eliminazione di un utente IAM nella Guida per l'utente IAM.
- Nella console IAM, rimuovi l'AccountAadminutente.

2. Accedi alla console IAM utilizzando le credenziali dell'account B. Elimina l'utente AccountBadmin.
3. Accedi alla console IAM utilizzando le credenziali dell'account C. Delete AccountCadmine l'utente Dave.

Risorse correlate

Per ulteriori informazioni relative a questa procedura dettagliata, consulta le seguenti risorse nella Guida per l'utente IAM:

- Creazione di un ruolo per delegare le autorizzazioni a un utente IAM
- Tutorial: delega l'accesso tramite i ruoli IAM Account AWS
- Gestione delle politiche IAM

In che modo Amazon S3 autorizza una richiesta

Quando Amazon S3 riceve una richiesta, ad esempio un'operazione su un bucket o su un oggetto, verifica innanzitutto che il richiedente disponga delle autorizzazioni necessarie. Amazon S3 valuta tutte le policy di accesso, le policy utente e le policy basate sulle risorse pertinenti (bucket policy, bucket access control list (ACL) e object ACL) per decidere se autorizzare la richiesta.

> **Note**
>
> Se il controllo delle autorizzazioni di Amazon S3 non riesce a trovare autorizzazioni valide, viene restituito un errore Access Denied (403 Forbidden) autorizzazione negata. Per ulteriori informazioni, consulta Risoluzione degli errori di accesso negato (403 proibito) in Amazon S3.

Per determinare se il richiedente è autorizzato a eseguire l'operazione specifica, Amazon S3 esegue le seguenti operazioni, nell'ordine, quando riceve una richiesta:

1. Converte tutte le policy di accesso pertinenti (policy utente, bucket policy e ACL) in fase di esecuzione in una serie di policy da valutare.
2. Valuta l'insieme di policy risultante nelle fasi successive. In ciascuna fase, Amazon S3 valuta un sottoinsieme di policy in un contesto specifico, in base all'autorità del contesto.
   a. Contesto dell'utente – Nel contesto dell'utente l'account padre a cui l'utente appartiene è l'autorità del contesto.

      Amazon S3 valuta un sottoinsieme di policy di proprietà dell'account padre. Questo sottoinsieme include la policy utente che l'account padre ha associato all'utente. Se il genitore possiede anche la risorsa nella richiesta (bucket o oggetto), Amazon S3 valuta anche le politiche delle risorse corrispondenti (bucket policy, bucket ACL e object ACL) contemporaneamente.

      Per eseguire l'operazione, un utente deve essere autorizzato da un account padre.

      Questa fase si applica solo se la richiesta viene eseguita da un utente in un Account AWS. Se la richiesta viene effettuata utilizzando le credenziali utente root di un Account AWS, Amazon S3 salta questo passaggio.
   b. Contesto del bucket: nel contesto del bucket, Amazon S3 valuta le politiche di proprietà del proprietario Account AWS del bucket.

Se la richiesta riguarda un'operazione su un bucket, il richiedente deve essere disporre dell'autorizzazione concessa dal proprietario del bucket. Se la richiesta riguarda un oggetto, Amazon S3 valuta tutte le policy appartenenti al proprietario del bucket per verificare che quest'ultimo non abbia negato in modo esplicito l'accesso all'oggetto. Se è stato impostato un rifiuto esplicito, Amazon S3 non autorizza la richiesta.

c. Contesto dell'oggetto – Se la richiesta riguarda un oggetto, Amazon S3 valuta il sottoinsieme di policy che appartengono al proprietario dell'oggetto.

Di seguito sono riportati alcuni scenari di esempio che illustrano come Amazon S3 autorizza una richiesta.

Example — Il richiedente è un preside IAM

Se il richiedente è un principale IAM, Amazon S3 deve determinare se il Account AWS genitore a cui appartiene il principale ha concesso l'autorizzazione principale necessaria per eseguire l'operazione. Inoltre, se la richiesta riguarda un'operazione su un bucket, ad esempio una richiesta per elencare il contenuto del bucket, Amazon S3 deve verificare che il proprietario del bucket abbia concesso al richiedente l'autorizzazione per eseguire l'operazione. Per eseguire un'operazione specifica su una risorsa, un principale IAM necessita dell'autorizzazione sia del genitore Account AWS a cui appartiene Account AWS sia del proprietario della risorsa.

Example — Il richiedente è un principale IAM: se la richiesta riguarda un'operazione su un oggetto che il proprietario del bucket non possiede

Se la richiesta riguarda un'operazione su un oggetto che il proprietario del bucket non possiede, oltre ad assicurarsi che il richiedente disponga delle autorizzazioni del proprietario dell'oggetto, Amazon S3 deve anche verificare la policy del bucket per assicurarsi che il proprietario del bucket non abbia impostato una negazione esplicita sull'oggetto. Il proprietario del bucket (che paga la fattura) può negare in modo esplicito l'accesso agli oggetti nel bucket, indipendentemente dall'utente a cui appartiene. Il proprietario del bucket può anche eliminare tutti gli oggetti nel bucket.

Per impostazione predefinita, quando un altro Account AWS carica un oggetto nel tuo bucket S3, quell'account (l'object writer) possiede l'oggetto, ha accesso ad esso e può concedere ad altri utenti l'accesso ad esso tramite gli elenchi di controllo degli accessi (ACL). È possibile utilizzare Object Ownership per modificare questo comportamento di default in modo che le ACL siano disabilitate e che tu, in qualità di proprietario del bucket, possieda automaticamente ogni oggetto nel tuo bucket.

Di conseguenza, il controllo degli accessi ai dati si basa su policy come le policy degli utenti IAM, le policy dei bucket S3, le policy degli endpoint del cloud privato virtuale (VPC) e le policy di controllo dei AWS Organizations servizi (SCP). Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

Per ulteriori informazioni su come Amazon S3 valuta le policy di accesso per autorizzare o negare le richieste di operazioni su bucket e oggetti, consulta i seguenti argomenti:

Argomenti

- In che modo Amazon S3 autorizza una richiesta per un'operazione su un bucket
- In che modo Amazon S3 autorizza una richiesta per un'operazione su un oggetto

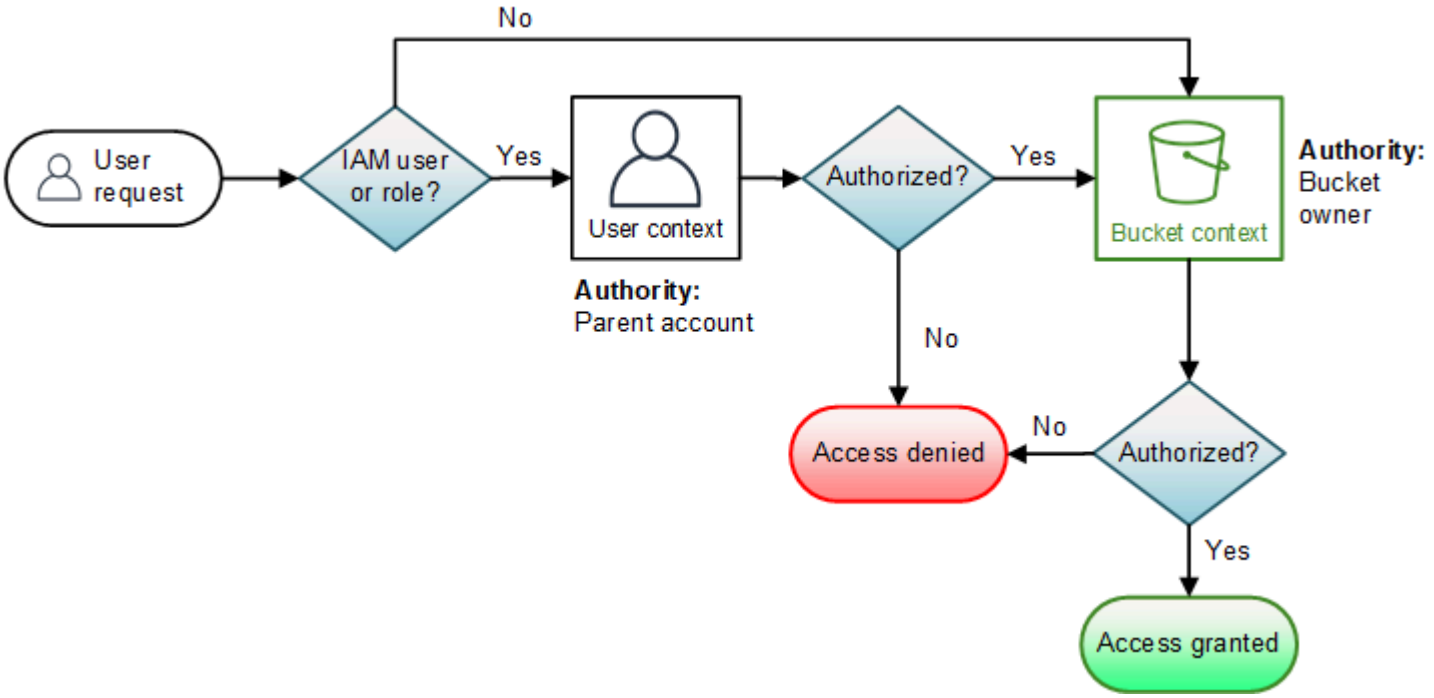
In che modo Amazon S3 autorizza una richiesta per un'operazione su un bucket

Quando Amazon S3 riceve una richiesta per un'operazione su un bucket, Amazon S3 converte tutte le autorizzazioni pertinenti in una serie di policy da valutare in fase di esecuzione. Le autorizzazioni pertinenti includono le autorizzazioni basate sulle risorse (ad esempio, le policy di bucket e le ACL dei bucket) e le policy utente se la richiesta proviene da un principale IAM. Amazon S3 valuta quindi il set di policy risultante in una serie di passaggi in base a un contesto specifico: contesto utente o contesto del bucket:

1. Contesto utente: se il richiedente è un titolare IAM, il principale deve avere l'autorizzazione del genitore a cui appartiene. Account AWS In questa fase, Amazon S3 valuta un sottoinsieme di policy appartenenti all'account padre (denominato anche autorità del contesto). Questo sottoinsieme include la policy utente che l'account padre ha associato al principale. Se l'account padre è anche proprietario della risorsa nella richiesta (in questo caso, il bucket), Amazon S3 valuta, allo stesso tempo, anche le policy della risorsa (la policy del bucket e l'ACL del bucket) corrispondenti. Ogni volta che viene eseguita una richiesta per un'operazione su un bucket, i log degli accessi del server registrano l'ID canonico del richiedente. Per ulteriori informazioni, consulta Registrazione delle richieste con registrazione dell'accesso al server.
2. Contesto del bucket – Il richiedente deve disporre delle autorizzazioni concesse dal proprietario del bucket per eseguire un'operazione specifica sul bucket. In questa fase, Amazon S3 valuta un sottoinsieme di policy di proprietà del proprietario del Account AWS bucket.

   Il proprietario del bucket può concedere l'autorizzazione utilizzando una policy del bucket o l'ACL del bucket. Se il Account AWS proprietario del bucket è anche l'account principale di un'entità principale IAM, può configurare le autorizzazioni del bucket in una policy utente.

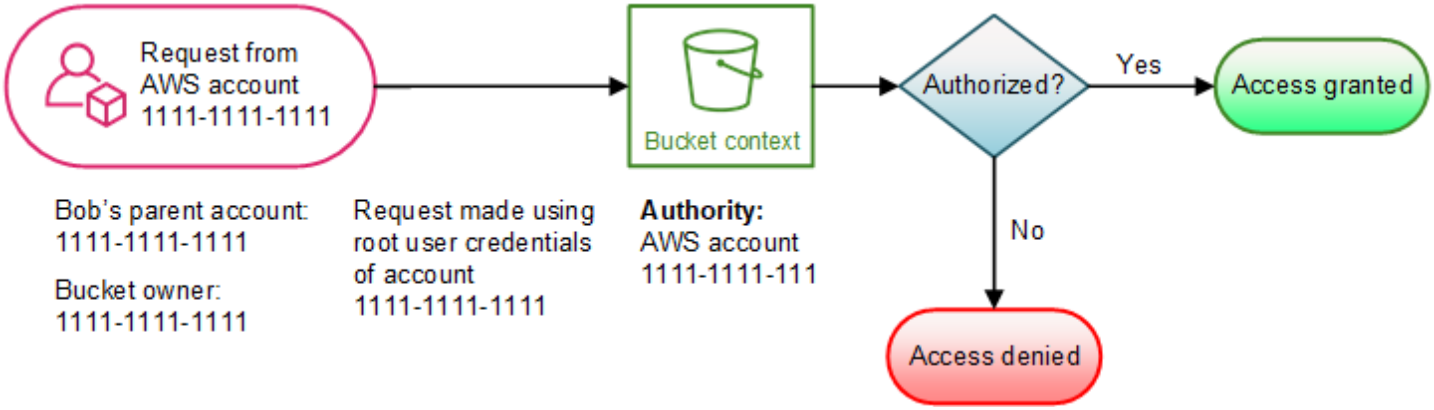
Di seguito è illustrato il grafico della valutazione basata sul contesto per un'operazione su un bucket.



Negli esempi seguenti viene illustrata la logica di valutazione.

Esempio 1: operazione su un bucket richiesta dal proprietario del bucket

In questo esempio, il proprietario del bucket invia una richiesta per un'operazione sul bucket utilizzando le credenziali dell'utente root dell' Account AWS.



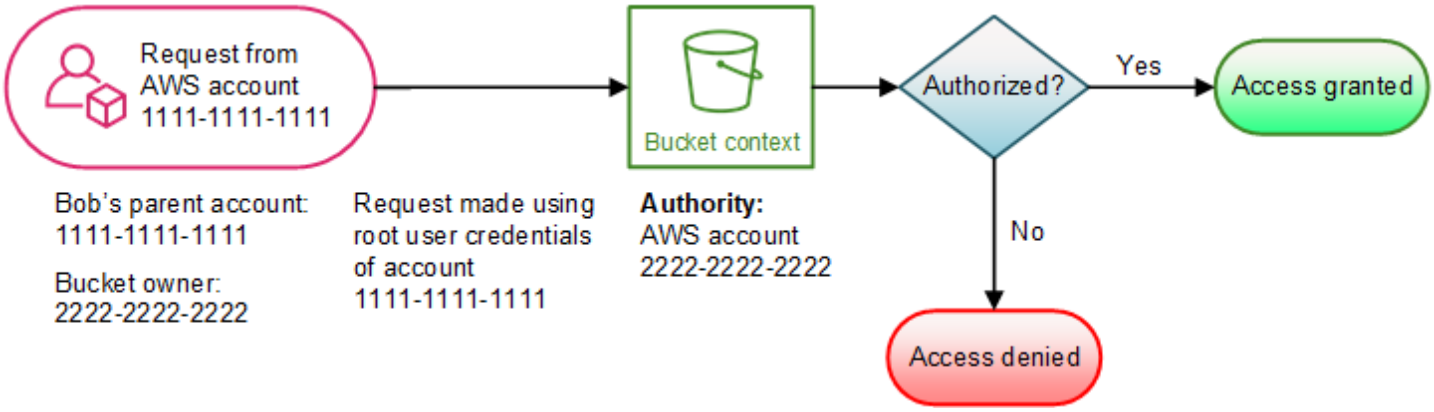
Amazon S3 esegue la valutazione del contesto come indicato di seguito:

1. Poiché la richiesta viene eseguita utilizzando le credenziali dell'utente root di un Account AWS, il contesto dell'utente non viene valutato.

2. Nel contesto del bucket, Amazon S3 esamina la policy del bucket per determinare se il richiedente dispone dell'autorizzazione per eseguire l'operazione. Amazon S3 autorizza la richiesta.

Esempio 2: operazione del bucket richiesta da un utente Account AWS che non è il proprietario del bucket

In questo esempio, viene eseguita una richiesta utilizzando le credenziali dell'utente root dell' Account AWS 1111-1111-1111 per un'operazione sul bucket che appartiene all' Account AWS 2222-2222-2222. Nessun utente IAM è coinvolto in questa richiesta.

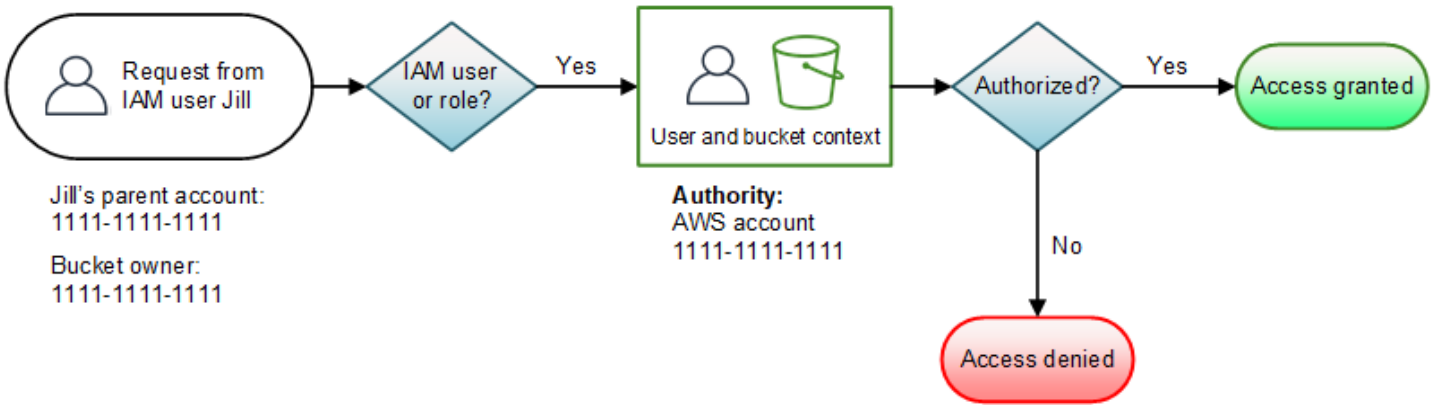


In questo esempio, Amazon S3 valuta il contesto come segue:

1. Poiché la richiesta viene effettuata utilizzando le credenziali dell'utente root di an Account AWS, il contesto utente non viene valutato.
2. Nel contesto del bucket, Amazon S3 esamina la policy del bucket. Se il proprietario del bucket (Account AWS 2222-2222-2222) non ha autorizzato Account AWS 1111-1111-1111 a eseguire l'operazione richiesta, Amazon S3 nega la richiesta. Altrimenti, Amazon S3 accetta la richiesta ed esegue l'operazione.

Esempio 3: operazione bucket richiesta da un principale IAM il cui genitore è anche il proprietario del bucket Account AWS

Nell'esempio, la richiesta viene inviata da Jill, un utente IAM nell' Account AWS 1111-1111-1111, che è anche il proprietario del bucket.



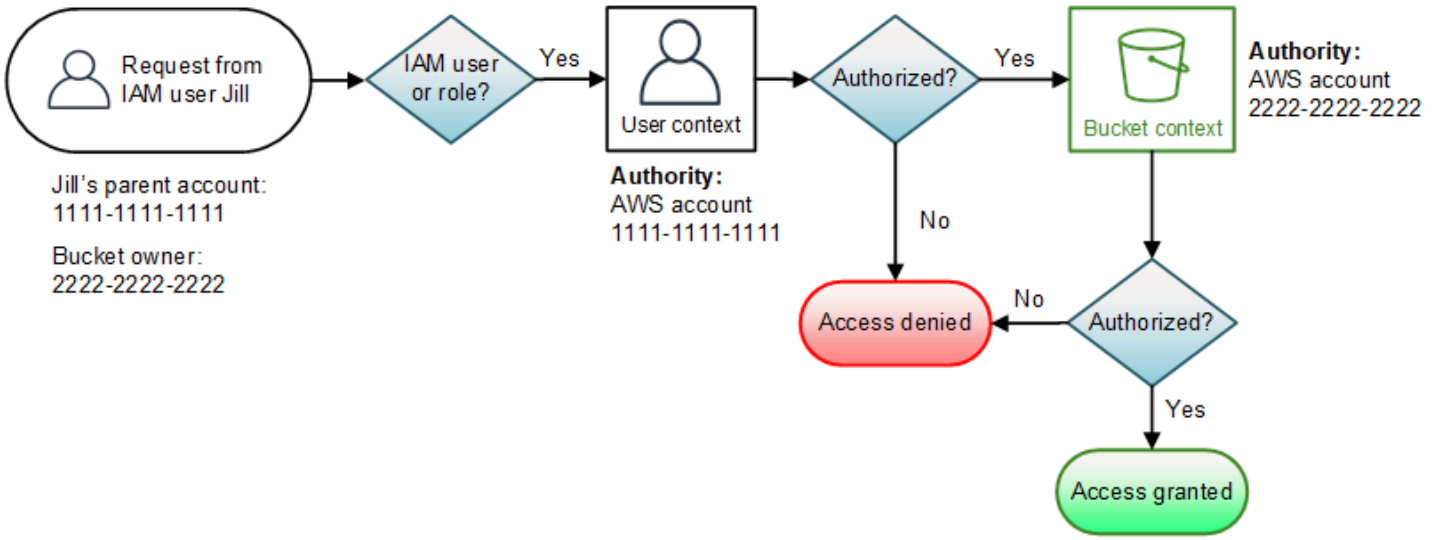
Amazon S3 esegue la seguente valutazione del contesto:

1. Poiché la richiesta proviene da un responsabile IAM, nel contesto dell'utente, Amazon S3 valuta tutte le policy che appartengono al principale per determinare se Jill è autorizzata Account AWS a eseguire l'operazione.

   In questo esempio, il genitore Account AWS 1111-1111-1111, a cui appartiene il principale, è anche il proprietario del bucket. Di conseguenza, oltre alla politica dell'utente, Amazon S3 valuta anche la policy del bucket e l'ACL del bucket nello stesso contesto perché appartengono allo stesso account.
2. Poiché Amazon S3 ha valutato la policy del bucket e l'ACL del bucket come parte del contesto dell'utente, non valuta il contesto del bucket.

Esempio 4: operazione bucket richiesta da un principale IAM il cui genitore non è il proprietario del bucket Account AWS

In questo esempio, la richiesta viene inviata da Jill, un utente IAM il cui genitore Account AWS è 1111-1111-1111, ma il bucket è di proprietà di un altro utente, 2222-2222-2222. Account AWS



Jill avrà bisogno delle autorizzazioni sia del genitore Account AWS che del proprietario del bucket. Amazon S3 valuta il contesto come indicato di seguito:

1. Poiché la richiesta proviene da un principale IAM, Amazon S3 valuta il contesto dell'utente esaminando le policy create dall'account per verificare che Jill disponga delle autorizzazioni necessarie. Se Jill è autorizzata, Amazon S3 passa alla valutazione del contesto del bucket. Se Jill non dispone dell'autorizzazione, nega la richiesta.
2. Nel contesto del bucket, Amazon S3 verifica che il proprietario del bucket 2222-2222-2222 abbia concesso a Jill (o al suo genitore) l'autorizzazione a eseguire l'operazione richiesta. Account AWS Se dispone di tale autorizzazione, Amazon S3 concede la richiesta ed esegue l'operazione. In caso contrario, Amazon S3 rifiuta la richiesta.

In che modo Amazon S3 autorizza una richiesta per un'operazione su un oggetto

Quando riceve una richiesta per un'operazione su un oggetto, Amazon S3 converte tutte le autorizzazioni rilevanti, ovvero le autorizzazioni basate sulle risorse (lista di controllo accessi (ACL) dell'oggetto, policy del bucket e ACL del bucket) e le policy utente IAM, in un set di policy da valutare in fase di esecuzione. Valuta quindi l'insieme di policy risultante in una serie di fasi. In ogni fase, valuta un sottoinsieme di politiche in tre contesti specifici: contesto utente, contesto del bucket e contesto dell'oggetto:

1. Contesto utente: se il richiedente è un principale IAM, il principale deve avere l'autorizzazione del genitore a cui appartiene. Account AWS In questa fase, Amazon S3 valuta un sottoinsieme di policy appartenenti all'account padre (denominato anche autorità del contesto). Questo sottoinsieme include la policy utente che l'account padre ha associato al principale. Se il

genitore possiede anche la risorsa nella richiesta (bucket o oggetto), Amazon S3 valuta contemporaneamente le politiche delle risorse corrispondenti (bucket policy, bucket ACL e object ACL).

> **Note**
>
> Se il genitore Account AWS possiede la risorsa (bucket o oggetto), può concedere le autorizzazioni relative alla risorsa al suo responsabile IAM utilizzando la politica dell'utente o la politica delle risorse.

2. Contesto del bucket: in questo contesto Amazon S3 valuta le policy che appartengono all' Account AWS proprietario del bucket.

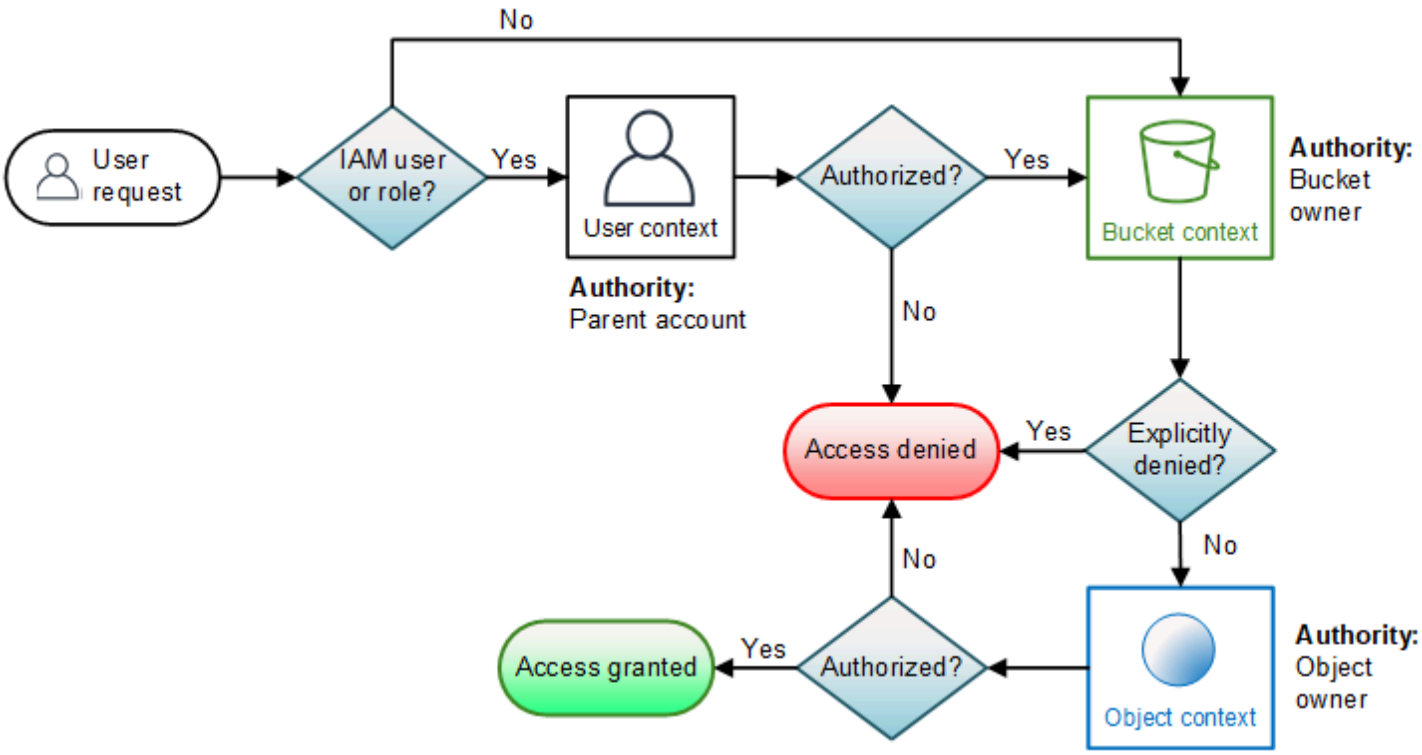
   Se il Account AWS proprietario dell'oggetto nella richiesta non è lo stesso del proprietario del bucket, Amazon S3 verifica le politiche se il proprietario del bucket ha negato esplicitamente l'accesso all'oggetto. Se è stato impostato un rifiuto esplicito sull'oggetto, Amazon S3 non autorizza la richiesta.

3. Contesto dell'oggetto – Il richiedente deve disporre delle autorizzazioni concesse dal proprietario dell'oggetto per eseguire un'operazione specifica sull'oggetto. In questa fase, Amazon S3 valuta l'ACL dell'oggetto.

> **Note**
>
> Se il proprietario del bucket corrisponde a quello dell'oggetto, l'accesso all'oggetto può essere concesso nella policy del bucket, che viene valutata nel contesto del bucket. Se i proprietari sono differenti, il proprietario dell'oggetto devono utilizzare un'ACL dell'oggetto per concedere le autorizzazioni. Se il Account AWS proprietario dell'oggetto è anche l'account principale a cui appartiene il principale IAM, può configurare le autorizzazioni dell'oggetto in una politica utente, che viene valutata nel contesto dell'utente. Per ulteriori informazioni su come utilizzare queste policy di accesso alternative, consulta la sezione Procedure dettagliate che utilizzano policy per gestire l'accesso alle risorse Amazon S3. Se in qualità di proprietario del bucket desideri possedere tutti gli oggetti nel tuo bucket e utilizzare politiche del bucket o politiche basate su IAM per gestire l'accesso a questi oggetti, puoi applicare l'impostazione imposta dal proprietario del bucket per Object Ownership. Con questa impostazione, in quanto proprietario del bucket possiedi automaticamente e hai il pieno controllo su ogni oggetto nel bucket. Le ACL del bucket e dell'oggetto non possono essere modificate e non sono più valutate per l'accesso. Per

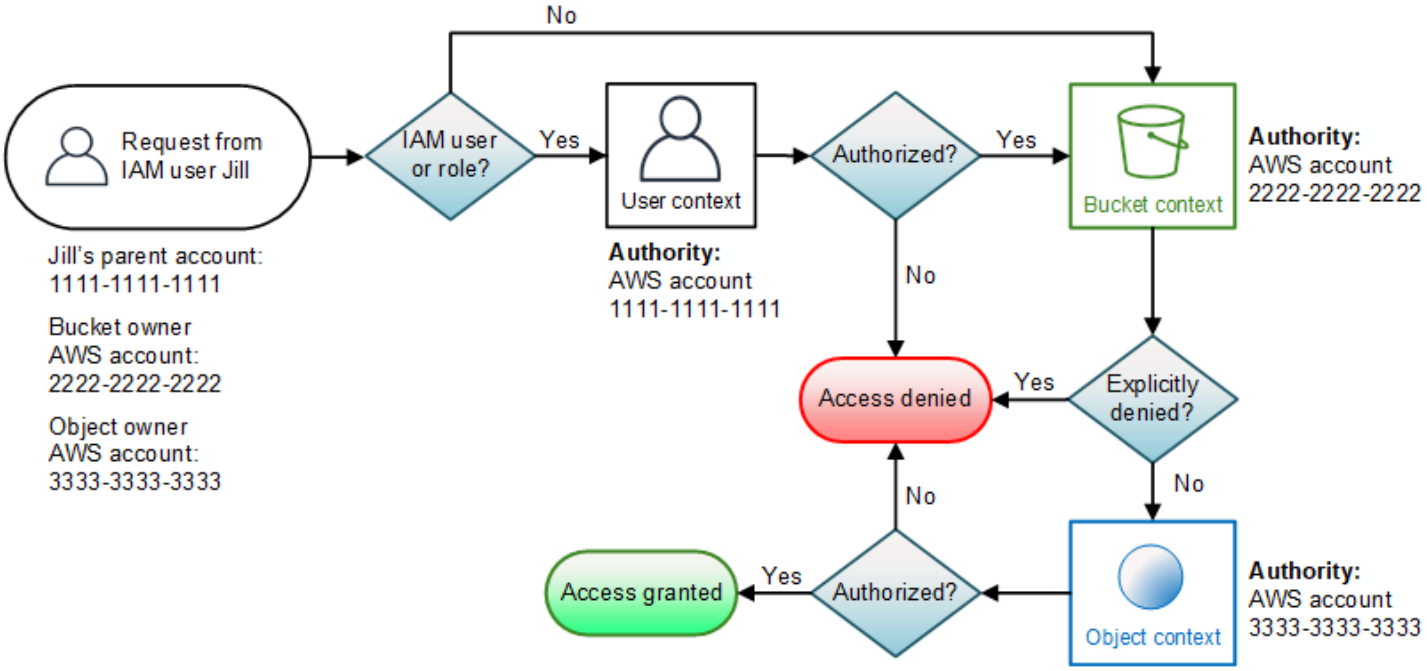
ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

Di seguito è riportata un'illustrazione della valutazione basata sul contesto per un'operazione su un oggetto.



Esempio di richiesta di funzionamento di un oggetto

In questo esempio, l'utente IAM Jill, il cui genitore Account AWS è 1111-1111-1111, invia una richiesta di operazione sull'oggetto (ad esempio,`GetObject`) per un oggetto di proprietà di Account AWS 3333-3333-3333 in un bucket di proprietà di 2222-2222-2222. Account AWS



Jill avrà bisogno dell'autorizzazione del genitore Account AWS, del proprietario del bucket e del proprietario dell'oggetto. Amazon S3 valuta il contesto come indicato di seguito:

1. Poiché la richiesta proviene da un principale IAM, Amazon S3 valuta il contesto dell'utente per verificare che il genitore Account AWS 1111-1111-1111 abbia concesso a Jill il permesso di eseguire l'operazione richiesta. Se Jill dispone di tale autorizzazione, Amazon S3 valuta il contesto del bucket. In caso contrario, Amazon S3 rifiuta la richiesta.
2. Nel contesto del bucket, il proprietario del bucket, 2222-2222-2222, è l'autorità del contesto. Account AWS Amazon S3 valuta la policy del bucket per determinare se il proprietario del bucket ha negato in modo esplicito l'accesso all'oggetto.
3. Nel contesto dell'oggetto, l'autorità del contesto è l' Account AWS 3333-3333-3333, ovvero il proprietario dell'oggetto. Amazon S3 valuta l'ACL dell'oggetto per determinare se Jill dispone dell'autorizzazione per accedere all'oggetto. In caso affermativo, Amazon S3 autorizza la richiesta.

AWS politiche gestite per Amazon S3

Una politica AWS gestita è una politica autonoma creata e amministrata da. AWS AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Ti consigliamo

pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i tuoi casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consultare Policy gestite da AWSnella Guida per l'utente di IAM.

AWS politica gestita: AmazonS3FullAccess

È possibile allegare la policy `AmazonS3FullAccess`alle identità IAM. Questa policy concede le autorizzazioni che consentono l'accesso completo ad Amazon S3.

Per visualizzare le autorizzazioni per questa policy, consulta AmazonS3FullAccess nella AWS Management Console.

AWS politica gestita: AmazonS3ReadOnlyAccess

È possibile allegare la policy `AmazonS3ReadOnlyAccess`alle identità IAM. Questa policy concede le autorizzazioni che consentono l'accesso in sola lettura ad Amazon S3.

Per visualizzare le autorizzazioni per questa policy, consulta AmazonS3ReadOnlyAccess nella AWS Management Console.

AWS Policy gestita: AmazonS3ObjectLambdaExecutionRolePolicy

Fornisce alle AWS Lambda funzioni le autorizzazioni necessarie per inviare dati a S3 Object Lambda quando vengono effettuate richieste a un punto di accesso S3 Object Lambda. Concede inoltre le autorizzazioni Lambda per la scrittura nei log di Amazon. CloudWatch

Per visualizzare le autorizzazioni per questa policy, consulta AmazonS3ObjectLambdaExecutionRolePolicy nella AWS Management Console.

Amazon S3 si aggiorna alle AWS politiche gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per Amazon S3 da quando questo servizio ha iniziato a tracciare queste modifiche.

Modifica	Descrizione	Data
Amazon S3 ha aggiunto le autorizzazioni Descrivi a `AmazonS3ReadOnlyAccess`	Amazon S3 ha aggiunto le autorizzazioni `s3:Describe*` a `AmazonS3ReadOnlyAccess` .	11 agosto 2023
Amazon S3 ha aggiunto le autorizzazioni per S3 Object Lambda `AmazonS3FullAccess` e `AmazonS3ReadOnlyAccess`	Amazon S3 ha aggiornato le policy `AmazonS3FullAccess` e `AmazonS3ReadOnlyAccess` in modo da includere le autorizzazioni per S3 Object Lambda.	27 settembre 2021
Amazon S3 ha aggiunto `AmazonS3ObjectLambdaExecutionRolePolicy`	Amazon S3 ha aggiunto una nuova policy AWS gestita chiamata che `AmazonS3ObjectLambdaExecutionRolePolicy` fornisce alle funzioni Lambda le autorizzazioni per interagire con S3 Object Lambda e scrivere nei log. CloudWatch	18 agosto 2021
Amazon S3 ha iniziato a tenere traccia delle modifiche	Amazon S3 ha iniziato a tracciare le modifiche per le sue politiche AWS gestite.	18 agosto 2021

Utilizzo dei ruoli collegati ai servizi per Amazon S3 Storage Lens

Per utilizzare Amazon S3 Storage Lens per raccogliere e aggregare i parametri su tutti i tuoi account AWS Organizations, devi innanzitutto assicurarti che per S3 Storage Lens sia abilitato l'accesso attendibile all'account di gestione della tua organizzazione. S3 Storage Lens crea un ruolo collegato al servizio (SLR) per consentirgli di ottenere l'elenco di appartenenza alla tua organizzazione. Account AWS Questo elenco di account viene utilizzato da S3 Storage Lens per raccogliere i parametri delle risorse S3 in tutti gli account membri quando il pannello di controllo o le configurazioni dello Storage Lens S3 vengono create o aggiornate.

Amazon S3 Storage Lens utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo univoco di ruolo IAM collegato direttamente a S3 Storage Lens. I ruoli collegati ai servizi sono predefiniti da S3 Storage Lens e includono tutte le autorizzazioni richieste dal servizio per chiamare altri utenti per tuo conto. Servizi AWS

Un ruolo collegato ai servizi semplifica la configurazione di S3 Storage Lens perché ti permette di evitare l'aggiunta manuale delle autorizzazioni necessarie. S3 Storage Lens definisce le autorizzazioni dei relativi ruoli associati ai servizi e, salvo diversamente definito, solo S3 Storage Lens potrà assumere i propri ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

È possibile eliminare il ruolo collegato ai servizi solo dopo avere eliminato le risorse correlate. Questa procedura protegge le risorse di S3 Storage Lens perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta Servizi AWS che funzionano con IAM e cerca i servizi che riportano Yes (Sì) nella colonna Service-Linked Role (Ruolo collegato al servizio). Scegli un link Yes (Sì) per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni di ruolo collegate ai servizi per Amazon S3 Storage Lens

S3 Storage Lens utilizza il ruolo collegato ai servizi denominato AWSServiceRoleForS3StorageLens: ciò consente l'accesso ai AWS servizi e alle risorse utilizzati o gestiti da S3 Storage Lens. Ciò consente a S3 Storage Lens di accedere alle risorse per tuo conto AWS Organizations .

Il ruolo collegato ai servizi S3 Storage Lens considera attendibile il seguente servizio nello storage dell'organizzazione:

- `storage-lens.s3.amazonaws.com`

La policy delle autorizzazioni del ruolo consente a S3 Storage Lens di eseguire le seguenti operazioni:

- `organizations:DescribeOrganization`

  `organizations:ListAccounts`

  `organizations:ListAWSServiceAccessForOrganization`

  `organizations:ListDelegatedAdministrators`

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi devi configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato ai servizi per S3 Storage Lens

Non devi creare manualmente un ruolo collegato ai servizi. Quando completi una delle seguenti attività mentre sei connesso agli account di AWS Organizations gestione o amministratore delegato, S3 Storage Lens crea automaticamente il ruolo collegato al servizio:

- Crea una configurazione del pannello di controllo S3 Storage Lens per la tua organizzazione nella console di Amazon S3.
- PUTuna configurazione S3 Storage Lens per la tua organizzazione che utilizza l'API REST e gli SDK. AWS CLI

> **Note**
>
> S3 Storage Lens supporterà un massimo di cinque amministratori delegati per organizzazione.

Se si elimina questo ruolo collegato ai servizi, le azioni precedenti lo ricreeranno all'occorrenza.

Esempio di policy per il ruolo collegato ai servizi S3 Storage Lens

Example Policy di autorizzazione per il ruolo collegato ai servizi S3 Storage Lens

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AwsOrgsAccess",
            "Effect": "Allow",
            "Action": [
                "organizations:DescribeOrganization",
                "organizations:ListAccounts",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:ListDelegatedAdministrators"
            ],
            "Resource": [
```

```
                "*"
            ]
        }
    ]
}
```

Modifica di un ruolo collegato ai servizi per Amazon S3 Storage Lens

S3 Storage Lens non consente di modificare il ruolo collegato al servizio. AWSServiceRoleForS3StorageLens Dopo aver creato un ruolo collegato al servizio, non puoi modificarne il nome, perché potrebbero farvi riferimento diverse entità. Puoi tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per Amazon S3 Storage Lens

Se non devi più utilizzare il ruolo collegato ai servizi, è consigliabile eliminarlo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato ai servizi prima di poterlo eliminare manualmente.

> **Note**
>
> Se il servizio Amazon S3 Storage Lens utilizza tale ruolo quando tenti di eliminare le risorse, è possibile che l'eliminazione non abbia esito positivo. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare il, AWSServiceRoleForS3StorageLens è necessario eliminare tutte le configurazioni di S3 Storage Lens a livello di organizzazione presenti in tutte Regioni AWS utilizzando gli account di AWS Organizations gestione o amministratore delegato.

Le risorse sono configurazioni S3 Storage Lens a livello di organizzazione. Usa S3 Storage Lens per pulire le risorse, quindi utilizza la console IAM, la CLI, l'API REST AWS o l'SDK per eliminare il ruolo.

Nell'API REST e negli SDK AWS CLI, è possibile scoprire le configurazioni di S3 Storage Lens `ListStorageLensConfigurations` in tutte le regioni in cui l'organizzazione ha creato configurazioni S3 Storage Lens. Utilizza l'azione `DeleteStorageLensConfiguration` per eliminare queste configurazioni in modo che sia possibile eliminare il ruolo.

> **Note**
>
> Per eliminare il ruolo collegato ai servizi, è necessario eliminare tutte le configurazioni S3 Storage Lens a livello di organizzazione in tutte le regioni in cui esistono.

Per eliminare le risorse di Amazon S3 Storage Lens utilizzate dalla reflex AWSServiceRoleForS3StorageLens

1. Per ottenere un elenco delle configurazioni a livello di organizzazione, è necessario utilizzare le configurazioni S3 Storage Lens `ListStorageLensConfigurations` in ogni regione in cui si dispone. Questo elenco può essere ottenuto anche dalla console Amazon S3.
2. Elimina queste configurazioni dagli endpoint regionali appropriati richiamando la chiamata `DeleteStorageLensConfiguration` API o utilizzando la console Amazon S3.

Per eliminare manualmente il ruolo collegato ai servizi mediante IAM

Dopo aver eliminato le configurazioni, elimina la AWSServiceRoleForS3StorageLens SLR dalla console IAM o richiamando l'API `DeleteServiceLinkedRole` IAM o utilizzando l'SDK o. AWS CLI AWS Per ulteriori informazioni, consulta Eliminazione del ruolo collegato ai servizi nella Guida per l'utente IAM.

Regioni supportate per i ruoli collegati ai servizi S3 Storage Lens

S3 Storage Lens supporta l'utilizzo di ruoli collegati al servizio in tutti i luoghi in cui il servizio è disponibile. Regioni AWS Per ulteriori informazioni, consulta la sezione Regioni ed endpoint di Amazon S3.

Risoluzione dei problemi di identità e accesso ad Amazon S3

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con Amazon S3 e IAM.

Argomenti

- Ho ricevuto un errore di accesso negato
- Non sono autorizzato a eseguire un'azione in Amazon S3
- Non sono autorizzato a eseguire iam: PassRole

- Desidero consentire a persone esterne Account AWS a me di accedere alle mie risorse Amazon S3

Ho ricevuto un errore di accesso negato

Verifica che non sia presente una `Deny` dichiarazione esplicita contro il richiedente a cui stai cercando di concedere le autorizzazioni nella policy del bucket o nella politica basata sull'identità.

Per informazioni dettagliate sulla risoluzione degli errori di accesso negato, consulta. Risoluzione dei problemi relativi agli errori di accesso negato (403 Accesso negato) in Amazon S3

Non sono autorizzato a eseguire un'azione in Amazon S3

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia ma non dispone di autorizzazioni `s3:`*`GetWidget`* fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 s3:GetWidget on resource: my-example-widget
```

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa *`my-example-widget`* utilizzando l'azione `s3:`*`GetWidget`*.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire l'`iam:PassRole`azione, le tue policy devono essere aggiornate per consentirti di trasferire un ruolo ad Amazon S3.

Alcuni Servizi AWS consentono di trasferire un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

Il seguente errore di esempio si verifica quando un utente IAM denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in Amazon S3. Tuttavia, l'azione richiede che il servizio

disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Desidero consentire a persone esterne Account AWS a me di accedere alle mie risorse Amazon S3

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per servizi che supportano policy basate su risorse o liste di controllo degli accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se Amazon S3 supporta queste funzionalità, consulta. Come funziona Amazon S3 con IAM
- Per scoprire come fornire l'accesso alle tue risorse su tutto Account AWS ciò che possiedi, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (Federazione delle identità) nella Guida per l'utente IAM.
- Per scoprire la differenza tra l'utilizzo di ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta Cross Account Resource Access in IAM nella IAM User Guide.

Gestione dell'accesso con S3 Access Grants

Per aderire al principio del privilegio minimo, definisci l'accesso granulare ai dati di Amazon S3 in base ad applicazioni, personaggi, gruppi o unità organizzative. Puoi utilizzare diversi approcci

per ottenere un accesso granulare ai tuoi dati in Amazon S3, a seconda della dimensione e della complessità dei modelli di accesso.

L'approccio più semplice per gestire l'accesso a small-to-medium numerosi set di dati in Amazon S3 AWS Identity and Access Management tramite i principali (IAM) consiste nel definire le policy di autorizzazione IAM e le policy dei bucket S3. Questa strategia funziona, a condizione che le policy necessarie rientrino nei limiti di dimensione delle policy del bucket S3 (20 KB) e delle policy IAM (5 KB), nonché nel numero di principali IAM consentiti per account.

Man mano che il numero di set di dati e di casi d'uso aumenta, potresti aver bisogno di più spazio per le policy. Un approccio che offre uno spazio significativamente maggiore per le istruzioni di policy consiste nell'utilizzare i Punti di accesso S3 come endpoint aggiuntivi per i bucket S3, poiché ogni punto di accesso può avere una propria policy. È possibile definire modelli di controllo degli accessi piuttosto granulari, poiché è possibile disporre di migliaia di punti di accesso Regione AWS per account, con una politica di dimensioni fino a 20 KB per ogni punto di accesso. Sebbene i Punti di accesso S3 aumentino la quantità di spazio disponibile per le policy, è necessario un meccanismo che consenta ai client di individuare il punto di accesso corretto per il set di dati corretto.

Un terzo approccio consiste nell'implementare un modello di broker di sessione IAM, in cui si implementa la logica di decisione di accesso e si generano dinamicamente credenziali di sessione IAM a breve termine per ogni sessione di accesso. Mentre l'approccio del broker di sessione IAM supporta arbitrariamente modelli di autorizzazioni dinamici nonché una scalabilità efficace, è necessario costruire la logica dei modelli di accesso.

Invece di utilizzare questi approcci, è possibile utilizzare S3 Access Grants per gestire l'accesso ai dati Amazon S3. S3 Access Grants fornisce un modello semplificato per definire le autorizzazioni di accesso ai dati in Amazon S3 per prefisso, bucket o oggetto. Inoltre, puoi utilizzare S3 Access Grants per concedere l'accesso sia ai principali IAM che direttamente a utenti o gruppi dalla tua directory aziendale.

In genere si definiscono le autorizzazioni per i dati in Amazon S3 mappando utenti e gruppi a set di dati. Puoi utilizzare S3 Access Grants per definire mappature di accesso diretto dei prefissi S3 a utenti e ruoli all'interno di bucket e oggetti Amazon S3. Con lo schema di accesso semplificato di S3 Access Grants, puoi concedere l'accesso in sola lettura, sola scrittura o lettura-scrittura in base al prefisso S3 sia ai principali IAM che direttamente a utenti o gruppi da una directory aziendale. Con queste funzionalità di S3 Access Grants, le applicazioni possono richiedere dati da Amazon S3 per conto dell'utente attualmente autenticato dell'applicazione.

Quando integri S3 Access Grants con la funzionalità di propagazione dell'identità affidabile di AWS IAM Identity Center, le tue applicazioni possono effettuare richieste Servizi AWS (incluso S3 Access Grants) direttamente per conto di un utente autenticato della directory aziendale. Le tue applicazioni non devono più mappare prima l'utente a un principale IAM. Inoltre, poiché le identità degli utenti finali vengono propagate fino ad Amazon S3, l'audit degli utenti che hanno avuto accesso a un determinato oggetto S3 è semplificato. Non è più necessario ricostruire la relazione tra diversi utenti e sessioni IAM. Quando utilizzi S3 Access Grants con la propagazione delle identità attendibili del Centro identità IAM, ogni evento relativo ai dati AWS CloudTrail per Amazon S3 contiene un riferimento diretto all'utente finale per conto del quale è stato effettuato l'accesso ai dati.

Per ulteriori informazioni sugli S3 Access Grants, consulta i seguenti argomenti.

Argomenti

Concetti di S3 Access Grants

S3 Access Grants introduce i seguenti concetti per il suo schema di accesso semplificato:

Istanze S3 Access Grants

> Un'istanza S3 Access Grants è un container logico per concessioni individuali che definiscono chi ha un determinato livello di accesso a un determinato tipo di dati Amazon S3. Puoi avere un'istanza S3 Access Grants per Regione AWS per Account AWS. Utilizzi questa istanza di S3 Access Grants per controllare l'accesso a tutti i bucket nello stesso account e. Regione AWS Se

desideri utilizzare S3 Access Grants per concedere l'accesso alle identità di utenti e gruppi nella directory aziendale, devi anche associare la tua istanza S3 Access Grants a un'istanza (IAM) Identity Center. AWS Identity and Access Management

Posizioni

Una posizione definisce il tipo di dati a cui l'istanza S3 Access Grants può concedere l'accesso. S3 Access Grants opera distribuendo credenziali IAM con accesso limitato a un particolare prefisso, bucket o oggetto S3. Alla posizione S3 Access Grants viene associato un ruolo IAM, dal quale vengono create queste sessioni temporanee. La configurazione di posizione più comune è una posizione singola in `s3://` per l'intera istanza S3 Access Grants, che può coprire l'accesso a tutti i bucket S3 dell'account e della Regione AWS. Puoi anche creare più posizioni nella tua istanza S3 Access Grants. Ad esempio, puoi registrare un bucket come una posizione `s3://`*`example-s3-bucket1`* per le concessioni che desideri limitare a questo bucket e puoi anche registrare la posizione `s3://` predefinita.

Concessioni

Per restringere l'ambito di accesso all'interno di una posizione, puoi creare concessioni individuali. Una concessione individuale in un'istanza S3 Access Grants consente a un'entità specifica, un principale IAM o un utente o un gruppo in una directory aziendale, di accedere a un prefisso, un bucket o un oggetto Amazon S3. Per ogni concessione, puoi definire un ambito (un prefisso, un bucket o un oggetto) e un livello di accesso (`READ`, `WRITE` o `READWRITE`) differenti. Ad esempio, potresti avere una concessione che consente a un particolare gruppo di directory aziendali di effettuare l'accesso `READ` `01234567-89ab-cdef-0123-456789abcdef` a `s3://`*`example-s3-bucket1`*`/projects/items/*`. Questa concessione consente agli utenti di quel gruppo di effettuare un accesso `READ` a ogni oggetto che ha un nome della chiave con il prefisso `projects/items/` nel bucket denominato *`example-s3-bucket1`*.

Credenziali temporanee di S3 Access Grants

Un'applicazione può richiedere le credenziali di just-in-time accesso chiamando una nuova operazione API S3 GetDataAccess, per richiedere l'accesso a un singolo oggetto, prefisso o bucket con un livello di autorizzazione di, o. `READ` `WRITE` `READWRITE` L'istanza S3 Access Grants valuta la richiesta `GetDataAccess` rispetto alle concessioni di cui dispone. Se esiste una concessione corrispondente, S3 Access Grants assume il ruolo IAM associato alla posizione della concessione corrispondente. S3 Access Grants assegna quindi le autorizzazioni della sessione IAM esattamente al bucket, al prefisso o all'oggetto S3 specificato dall'ambito della concessione. L'ora di scadenza delle credenziali di accesso temporanee è predefinita a 1 ora, ma è possibile impostarla su qualsiasi valore compreso tra 15 minuti e 12 ore.

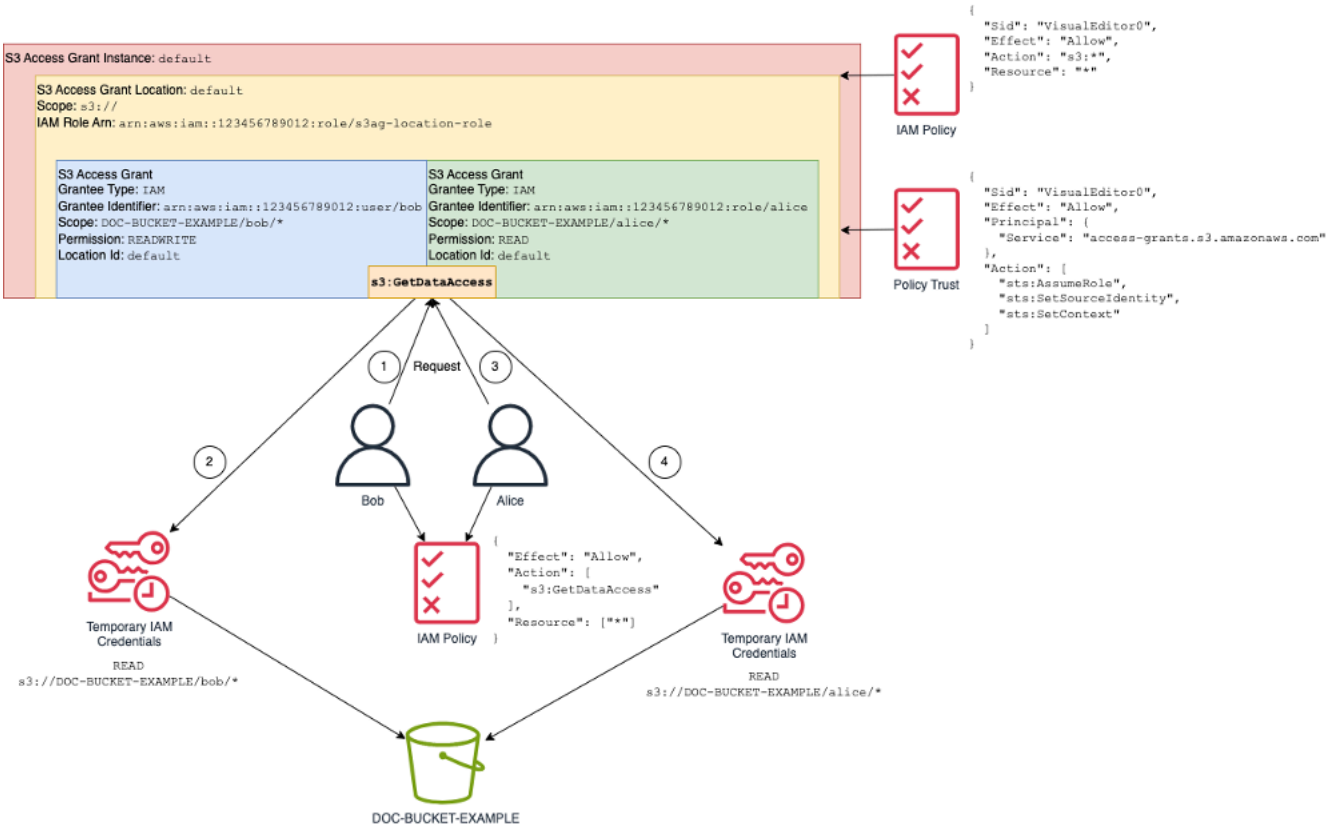
Come funziona

Nel diagramma seguente, una posizione Amazon S3 predefinita con l'ambito `s3://` è registrata con il ruolo IAM `s3ag-location-role`. Questo ruolo IAM dispone delle autorizzazioni per eseguire azioni Amazon S3 all'interno dell'account quando le sue credenziali vengono ottenute tramite S3 Access Grants.

In questa posizione, vengono create due concessioni di accesso individuali per due utenti IAM. All'utente IAM Bob vengono concessi gli accessi `READ` e `WRITE` al prefisso `bob/` nel bucket `DOC-BUCKET-EXAMPLE`. A un altro ruolo IAM, Alice, viene concesso solo `READ` l'accesso al `alice/` prefisso nel bucket. `DOC-BUCKET-EXAMPLE` Viene definita una concessione, colorata in blu, per consentire a Bob di accedere al prefisso `bob/` nel bucket `DOC-BUCKET-EXAMPLE`. Viene definita una concessione, colorata in verde, per consentire ad Alice di accedere al prefisso `alice/` nel bucket `DOC-BUCKET-EXAMPLE`.

Quando Bob è il momento di passare ai `READ` dati, il ruolo IAM associato alla posizione in cui si trova la sua concessione chiama l'operazione dell'API S3 Access GetDataAccessGrants. Se Bob tenta di leggere (`READ`) un qualsiasi prefisso o oggetto S3 che inizia con `s3://DOC-BUCKET-EXAMPLE/bob/*`, la richiesta `GetDataAccess` restituisce un set di credenziali di sessione IAM temporanee con autorizzazione a `s3://DOC-BUCKET-EXAMPLE/bob/*`. Allo stesso modo, Bob può scrivere (`WRITE`) su qualsiasi prefisso o oggetto S3 che inizi con `s3://DOC-BUCKET-EXAMPLE/bob/*`, perché anche la concessione lo consente.

Alice, invece, può leggere (`READ`) tutto ciò che inizia con `s3://DOC-BUCKET-EXAMPLE/alice/`. Tuttavia, se prova a scrivere (`WRITE`) qualunque cosa su un qualsiasi bucket, prefisso o oggetto in `s3://`, riceverà un errore Accesso negato (403), perché non esiste alcuna concessione che le dia accesso in scrittura (`WRITE`) ai dati. Inoltre, se Alice richiede un qualsiasi livello di accesso (`READ` o `WRITE`) ai dati esterni a`s3://DOC-BUCKET-EXAMPLE/alice/`, riceverà nuovamente un errore di accesso negato.



Questo modello si adatta a un numero elevato di utenti e bucket e semplifica la gestione di tali autorizzazioni. Anziché modificare le policy dei bucket S3 potenzialmente grandi ogni volta che desideri aggiungere o rimuovere una relazione di accesso prefisso-utente individuale, puoi aggiungere e rimuovere concessioni individuali e discrete.

S3 Access Grants e identità delle directory aziendali

Puoi utilizzare Amazon S3 Access Grants per concedere l'accesso ai principali AWS Identity and Access Management (utenti o ruoli) (IAM), sia nello stesso Account AWS che in altri. Tuttavia, in molti casi, l'entità che accede ai dati è un utente finale della directory aziendale. Invece di concedere l'accesso ai principali IAM, puoi utilizzare S3 Access Grants per concedere l'accesso direttamente agli utenti e ai gruppi aziendali. Con S3 Access Grants, non è più necessario mappare le identità aziendali a principali IAM intermedi per accedere ai dati S3 tramite le applicazioni aziendali.

Questa nuova funzionalità, il supporto per l'utilizzo delle identità degli utenti finali per l'accesso ai dati, viene fornita associando l'istanza S3 Access Grants a un'istanza. AWS IAM Identity Center IAM Identity Center supporta provider di identità basati su standard ed è l'hub di tutti i servizi o funzionalità, inclusi S3 Access Grants, che supportano le identità degli utenti finali AWS . Il Centro identità IAM fornisce supporto per l'autenticazione delle identità aziendali attraverso la sua

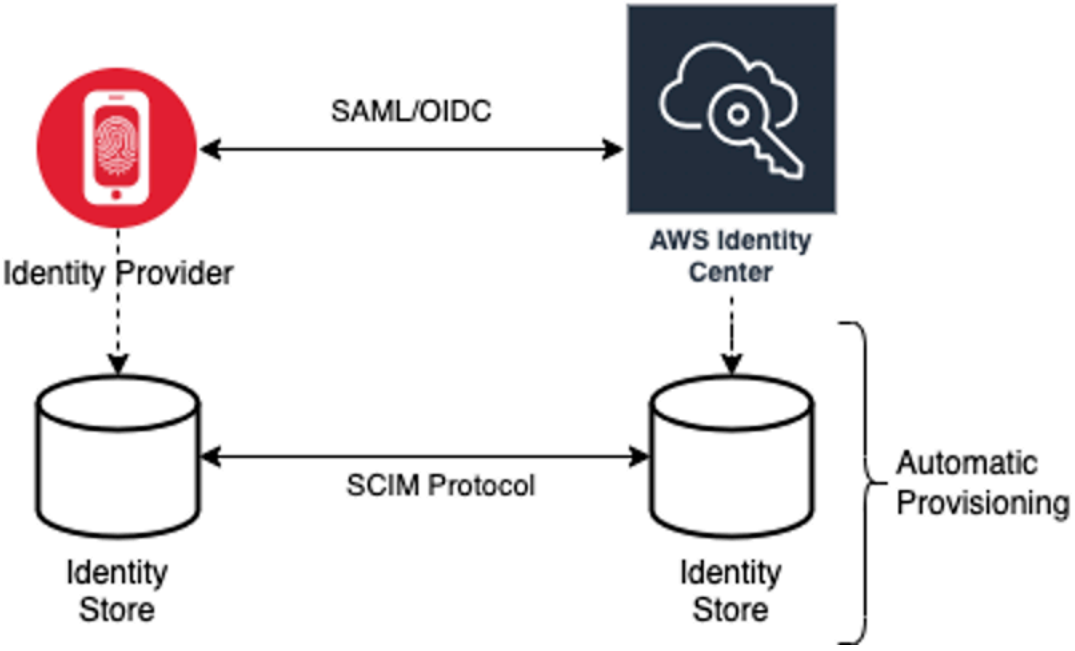
funzionalità Propagazione delle identità attendibili. Per ulteriori informazioni, consulta la pagina Propagazione delle identità attendibili tra le applicazioni.

Prima di iniziare con il supporto delle identità della forza lavoro in S3 Access Grants, devi configurare il provisioning delle identità tra il tuo gestore delle identità aziendali e il Centro identità IAM, come prerequisito. Il Centro identità IAM supporta gestori delle identità aziendali come Okta, Microsoft Entra ID (in precedenza Azure Active Directory) o qualsiasi altro gestore dell'identità digitale (IdP) esterno che supporti il protocollo System for Cross-domain Identity Management (SCIM). Quando connetti il Centro identità IAM al tuo gestore dell'identità digitale (IdP) e abiliti il provisioning automatico, gli utenti e i gruppi del tuo IdP vengono sincronizzati nell'archivio di identità nel Centro identità IAM. Dopo questo passaggio, IAM Identity Center ha una propria visione degli utenti e dei gruppi, in modo che tu possa fare riferimento a loro utilizzando altre Servizi AWS funzionalità, come S3 Access Grants. Per ulteriori informazioni sulla configurazione del provisioning automatico del Centro identità IAM, consulta la sezione Provisioning automatico nella Guida per l'utente di AWS IAM Identity Center .

IAM Identity Center è integrato AWS Organizations in modo da poter gestire centralmente le autorizzazioni su più account Account AWS senza configurare manualmente ciascuno dei tuoi account. In un'organizzazione tipica, l'amministratore delle identità configura un'istanza del Centro identità IAM per l'intera organizzazione, come unico punto di sincronizzazione delle identità. Questa istanza di IAM Identity Center viene in genere eseguita in un ambiente dedicato Account AWS dell'organizzazione. In questa configurazione comune, puoi fare riferimento alle identità di utenti e gruppi in S3 Access Grants da qualsiasi Account AWS parte dell'organizzazione.

Tuttavia, se AWS Organizations l'amministratore non ha ancora configurato un'istanza centrale di IAM Identity Center, puoi crearne una locale nello stesso account dell'istanza S3 Access Grants. Tale configurazione è più comune per i nostri casi proof-of-concept d'uso di sviluppo locale. In tutti i casi, l'istanza IAM Identity Center deve essere la Regione AWS stessa dell'istanza S3 Access Grants a cui verrà associata.

Nel diagramma seguente di una configurazione del Centro identità IAM con un gestore dell'identità digitale (IdP) esterno, l'IdP è configurato con SCIM per sincronizzare l'archivio di identità dal gestore dell'IdP all'archivio di identità nel Centro identità IAM.



Per utilizzare le identità delle directory aziendali con S3 Access Grants, procedi come segue:

- Configura il provisioning automatico nel Centro identità IAM per sincronizzare le informazioni su utenti e gruppi dal tuo gestore dell'identità digitale (IdP) nel Centro identità IAM.
- Configura l'origine di identità esterna nel Centro identità IAM come emittente del token affidabile. Per ulteriori informazioni, consulta la pagina Propagazione delle identità attendibili tra le applicazioni nella Guida per l'utente di AWS IAM Identity Center .
- Associa l'istanza S3 Access Grants all'istanza del Centro identità IAM. Puoi farlo quando crei la tua istanza S3 Access Grants. Se hai già creato la tua istanza S3 Access Grants, consulta Associazione o annullamento dell'associazione dell'istanza del Centro identità IAM.

In che modo le identità delle directory possono accedere ai dati S3

Supponiamo di avere utenti della directory aziendale che devono accedere ai dati S3 attraverso un'applicazione aziendale, ad esempio un'applicazione per la visualizzazione di documenti, che è integrata con il gestore dell'identità digitale (IdP) esterno (ad esempio, Okta) per autenticare gli utenti. L'autenticazione dell'utente in queste applicazioni viene in genere effettuata tramite reindirizzamenti nel browser web dell'utente. Poiché gli utenti presenti nella directory non sono principali IAM, l'applicazione necessita di credenziali IAM con cui richiamare l'operazione API `GetDataAccess` S3 Access Grants per ottenere le credenziali di accesso ai dati S3 per conto degli utenti. A differenza degli utenti e dei ruoli IAM che ottengono le credenziali da soli, l'applicazione necessita di un modo

per rappresentare un utente della directory, che non è mappato a un ruolo IAM, in modo che l'utente possa accedere ai dati tramite S3 Access Grants.

Questa transizione, da utente di directory autenticato a chiamante IAM in grado di effettuare richieste a S3 Access Grants per conto dell'utente della directory, viene effettuata dall'applicazione tramite la funzionalità Emittente del token affidabile del Centro identità IAM. L'applicazione, dopo aver autenticato l'utente della directory, dispone di un token di identità dell'IdP (ad esempio, Okta) che rappresenta l'utente della directory in base a Okta. La configurazione dell'emittente del token affidabile nel Centro identità IAM consente all'applicazione di scambiare questo token Okta (il tenant Okta è configurato come "emittente attendibile") con un token di identità diverso dal Centro identità IAM che rappresenterà in modo sicuro l'utente della directory all'interno dei Servizi AWS. L'applicazione dati assumerà quindi un ruolo IAM, fornendo il token dell'utente della directory proveniente dal Centro identità IAM come contesto aggiuntivo. L'applicazione può utilizzare la sessione IAM risultante per chiamare S3 Access Grants. Il token rappresenta sia l'identità dell'applicazione (il principale IAM stesso) sia l'identità dell'utente della directory.

Il passaggio principale di questa transizione è lo scambio di token. L'applicazione esegue questo scambio di token chiamando l'operazione API `CreateTokenWithIAM` nel Centro identità IAM. Naturalmente, anche questa è una chiamata AWS API e richiede che un preside IAM la firmi. Il principale IAM che effettua questa richiesta è in genere un ruolo IAM associato all'applicazione. Ad esempio, se l'applicazione viene eseguita su Amazon EC2, la richiesta `CreateTokenWithIAM` viene in genere eseguita dal ruolo IAM associato all'istanza EC2 su cui viene eseguita l'applicazione. Il risultato di una `CreateTokenWithIAM` chiamata riuscita è un nuovo token di identità, che verrà riconosciuto all'interno Servizi AWS.

Il passaggio successivo, prima che l'applicazione possa chiamare `GetDataAccess` per conto dell'utente della directory, prevede che l'applicazione ottenga una sessione IAM che includa l'identità dell'utente della directory. L'applicazione esegue questa operazione con una `AssumeRole` richiesta AWS Security Token Service (AWS STS) che include anche il token IAM Identity Center per l'utente della directory come contesto di identità aggiuntivo. Questo contesto aggiuntivo consente al Centro identità IAM di propagare l'identità dell'utente della directory alla fase successiva. Il ruolo IAM assunto dall'applicazione è il ruolo che necessiterà delle autorizzazioni IAM per chiamare l'operazione `GetDataAccess`.

Dopo aver assunto il ruolo IAM di portatore di identità con il token del Centro identità IAM per l'utente della directory come contesto aggiuntivo, l'applicazione dispone ora di tutti gli elementi necessari per inviare una richiesta firmata a `GetDataAccess` per conto dell'utente della directory autenticato.

La propagazione dei token si basa sui seguenti passaggi:

Creazione di un'applicazione del Centro identità IAM

Innanzitutto, crea una nuova applicazione nel Centro identità IAM. Questa applicazione utilizzerà un modello che consente al Centro identità IAM di identificare il tipo di impostazioni dell'applicazione che è possibile utilizzare. Il comando per creare l'applicazione richiede di fornire il nome della risorsa Amazon (ARN) dell'istanza del Centro identità IAM, un nome di applicazione e il nome della risorsa Amazon (ARN) del provider dell'applicazione. Il provider dell'applicazione è il provider dell'applicazione SAML o OAuth che l'applicazione utilizzerà per effettuare le chiamate al Centro identità IAM.

Per utilizzare il seguente comando di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

```
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
```

Risposta:

```
{
   "ApplicationArn": "arn:aws:sso::123456789012:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d"
}
```

Creazione di un emittente del token affidabile

Ora che hai la tua applicazione del Centro identità IAM, il passaggio successivo consiste nel configurare un emittente del token affidabile che verrà utilizzato per scambiare i valori `IdToken` del tuo gestore dell'identità digitale con i token del Centro identità IAM. Per completare questa fase, sono necessari i seguenti elementi:

- L'URL dell'emittente del gestore dell'identità
- Il nome emittente del token affidabile
- Il percorso dell'attributo claim
- Il percorso dell'attributo identity store
- L'opzione di recupero JSON Web Key Set (JWKS)

Il percorso dell'attributo claim è l'attributo del gestore delle identità che verrà utilizzato per mappare l'attributo identity store. Normalmente, il percorso dell'attributo claim è l'indirizzo e-mail dell'utente, ma è possibile utilizzare altri attributi per eseguire la mappatura.

Crea un file denominato `oidc-configuration.json` con le informazioni seguenti. Per utilizzare questo file, sostituisci *`user input placeholders`* con le tue informazioni specifiche.

```
{
  "OidcJwtConfiguration":
     {
      "IssuerUrl": "https://login.microsoftonline.com/a1b2c3d4-abcd-1234-b7d5-
b154440ac123/v2.0",
      "ClaimAttributePath": "preferred_username",
      "IdentityStoreAttributePath": "userName",
      "JwksRetrievalOption": "OPEN_ID_DISCOVERY"
     }
}
```

Per creare l'emittente del token affidabile, esegui questo comando. Per utilizzare questo comando di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

```
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
```

Risposta

```
{
  "TrustedTokenIssuerArn": "arn:aws:sso::123456789012:trustedTokenIssuer/
ssoins-1234567890abcdef/tti-43b4a822-1234-1234-1234-a1b2c3d41234"
}
```

Connessione dell'applicazione del Centro identità IAM con l'emittente del token affidabile

L'emittente del token affidabile richiede alcune altre impostazioni di configurazione per funzionare. Imposta i destinatari di cui si fiderà l'emittente del token affidabile. I destinatari rappresentano il valore all'interno di `IdToken` che è identificato dalla chiave e può essere trovato nelle impostazioni del gestore dell'identità. Per esempio:

```
1234973b-abcd-1234-abcd-345c5a9c1234
```

Crea un file denominato `grant.json` che abbia il seguente contenuto. Per utilizzare questo file, modifica i destinatari in modo che corrispondano alle impostazioni del tuo gestore dell'identità e fornisci il nome della risorsa Amazon (ARN) dell'emittente del token affidabile che è stato restituito dal comando precedente.

```
{
   "JwtBearer":
     {
       "AuthorizedTokenIssuers":
         [
           {
             "TrustedTokenIssuerArn": "arn:aws:sso::123456789012:trustedTokenIssuer/
ssoins-1234567890abcdef/tti-43b4a822-1234-1234-1234-a1b2c3d41234",
               "AuthorizedAudiences":
                 [
                   "1234973b-abcd-1234-abcd-345c5a9c1234"
                 ]
           }
         ]
     }
 }
```

Esegui il seguente comando di esempio. Per utilizzare questo comando, sostituisci *`user input placeholders`* con le tue specifiche informazioni.

```
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 \
```

Questo comando configura l'emittente del token affidabile in modo che consideri attendibili i destinatari presenti nel file `grant.json` e li colleghi all'applicazione creata nel primo passaggio per lo scambio di token del tipo `jwt-bearer`. La stringa `urn:ietf:params:oauth:grant-type:jwt-bearer` non è una stringa arbitraria. È uno spazio dei nomi registrato nei profili di asserzione OAuth JSON Web Token (JWT). Puoi trovare ulteriori informazioni su questo spazio dei nomi in RFC 7523.

Successivamente, utilizza il seguente comando per impostare gli ambiti che l'emittente del token affidabile includerà nello scambio dei valori `IdToken` dal tuo provider dell'identità. Per S3 Access Grants, il valore del parametro `--scope` è `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"
```

L'ultimo passaggio consiste nell'allegare una policy delle risorse all'applicazione del Centro identità IAM. Questa policy consentirà al ruolo IAM dell'applicazione di effettuare richieste all'operazione API `sso-oauth:CreateTokenWithIAM` e ricevere i valori `IdToken` dal Centro identità IAM.

Crea un file denominato `authentication-method.json` che abbia il seguente contenuto. Sostituisci *123456789012* con l'ID del tuo account.

```
{
   "Iam":
       {
         "ActorPolicy":
             {
                "Version": "2012-10-17",
                    "Statement":
                    [
                        {
                           "Effect": "Allow",
                            "Principal":
                            {
                              "AWS": "arn:aws:iam::123456789012:role/webapp"
                            },
                           "Action": "sso-oauth:CreateTokenWithIAM",
                            "Resource": "*"
                        }
                    ]
                }
            }
        }
```

Per collegare la policy all'applicazione del Centro identità IAM, esegui il 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
```

Questo completa le impostazioni di configurazione per l'utilizzo di S3 Access Grants con gli utenti della directory tramite un'applicazione web. Puoi testare questa configurazione direttamente nell'applicazione oppure puoi chiamare l'operazione API `CreateTokenWithIAM` utilizzando il seguente comando da un ruolo IAM consentito nella policy dell'applicazione del Centro identità 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 risposta sarà simile alla seguente:

```
{
    "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"
    ]
}
```

Se decodifichi il valore `IdToken` codificato con base64, puoi vedere le coppie chiave-valore in formato JSON. La chiave `sts:identity_context` contiene il valore che l'applicazione deve inviare nella richiesta `sts:AssumeRole` per includere le informazioni sull'identità dell'utente della directory. Di seguito è riportato un esempio di `IdToken` decodificato:

```
{
    "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
}
```

Puoi ottenere il valore da `sts:identity_context` e trasmettere queste informazioni in una chiamata `sts:AssumeRole`. Di seguito è riportato un esempio di sintassi CLI. Il ruolo da assumere è un ruolo temporaneo con autorizzazioni per richiamare `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"
```

Ora puoi utilizzare le credenziali ricevute da questa chiamata per richiamare l'operazione API `s3:GetDataAccess` e ricevere le credenziali finali con accesso alle risorse S3.

Nozioni di base su S3 Access Grants

Amazon S3 Access Grants è una funzionalità di Amazon S3 che fornisce una soluzione scalabile di controllo degli accessi per i dati S3. S3 Access Grants è un fornitore di credenziali S3, il che significa che devi registrare su S3 Access Grants il tuo elenco di concessioni e specificarne il livello. Successivamente, quando gli utenti o i client devono accedere ai tuoi dati S3, devono prima chiedere le credenziali a S3 Access Grants. Se esiste una concessione corrispondente che autorizza l'accesso, S3 Access Grants fornisce credenziali di accesso temporanee con privilegi minimi. Gli utenti o i client possono quindi utilizzare le credenziali fornite da S3 Access Grants per accedere ai dati S3. Tenendo presente questo, se i requisiti relativi ai dati S3 richiedono una configurazione di

autorizzazioni complessa o di grandi dimensioni, puoi utilizzare S3 Access Grants per dimensionare le autorizzazioni relative ai dati S3 per utenti, gruppi, ruoli e applicazioni.

Nella maggior parte dei casi d'uso, puoi gestire il controllo degli accessi per i tuoi dati S3 utilizzando AWS Identity and Access Management (IAM) con bucket policy o policy basate sull'identità IAM.

Tuttavia, se hai requisiti di controllo degli accessi S3 complessi, come i seguenti, puoi ottenere grandi vantaggi dall'utilizzo di S3 Access Grants:

- Stai per raggiungere il limite di dimensioni della policy del bucket di 20 KB.
- Concedi alle identità umane, ad esempio agli utenti e ai gruppi Microsoft Entra ID (in precedenza Azure Active Directory), Okta o Ping, l'accesso ai dati S3 per analisi e big data.
- Devi fornire l'accesso multi-account senza apportare aggiornamenti frequenti alle policy IAM.
- I tuoi dati non sono strutturati e sono a livello di oggetto anziché strutturati, in formato riga e colonna.

Di seguito è riportato il flusso di lavoro di S3 Access Grants:

Fasi	Descrizione
1	Creazione di un'istanza S3 Access Grants Per iniziare, avvia un'istanza S3 Access Grants che conterrà le tue concessioni di accesso individuali.
2	Registrazione di una posizione In secondo luogo, registra una posizione dati S3 (ad esempio quella predefinita `s3://`), quindi specifica un ruolo IAM predefinito che S3 Access Grants assume quando fornisce l'accesso alla posizione dei dati S3. Puoi anche aggiungere posizioni personalizzate a bucket o prefissi specifici e mapparle a ruoli IAM personalizzati.
3	Creazione di concessioni Crea concessioni di autorizzazione individuali. In queste concessioni di autorizzazioni devi specificare la posizione S3

Fasi	Descrizione
	registrata, l'ambito di accesso ai dati all'interno della posizione , l'identità dell'assegnatario e il suo livello di accesso (`READ`, `WRITE` o `READWRITE`).
4	Richiesta di accesso ai dati S3 Quando gli utenti, le applicazioni e gli utenti Servizi AWS desiderano accedere ai dati S3, devono prima effettuare una richiesta di accesso. S3 Access Grants determina se la richiesta deve essere autorizzata. Se esiste una concessione corrispon dente che autorizza l'accesso, S3 Access Grants utilizza il ruolo IAM della posizione registrata associato a tale concessione per fornire le credenziali temporanee al richiedente.
5	Accesso ai dati S3 Le applicazioni utilizzano le credenziali temporanee fornite da S3 Access Grants per accedere ai dati S3.

Creazione di un'istanza S3 Access Grants

Per iniziare a usare Amazon S3 Access Grants, devi prima creare un'istanza S3 Access Grants. Puoi creare solo un'istanza S3 Access Grants per account. Regione AWS L'istanza S3 Access Grants funge da container per le risorse S3 Access Grants, che includono concessioni e posizioni registrate.

Con S3 Access Grants, puoi creare concessioni di autorizzazione ai tuoi dati S3 per utenti e ruoli AWS Identity and Access Management (IAM). Se hai aggiunto la tua directory di identità aziendale a AWS IAM Identity Center, puoi associare questa istanza IAM Identity Center della tua directory aziendale alla tua istanza S3 Access Grants. Quindi, puoi creare concessioni di accesso per gli utenti e i gruppi aziendali. Se non hai ancora aggiunto la tua directory di identità aziendale al Centro identità IAM, puoi associare l'istanza S3 Access Grants a un'istanza del Centro identità IAM in un secondo momento.

Puoi creare un'istanza S3 Access Grants utilizzando la console Amazon S3,AWS CLI(), AWS Command Line Interface l'API REST di Amazon S3 e gli SDK. AWS

Utilizzo della console S3

Prima di poter concedere l'accesso ai tuoi dati S3 con S3 Access Grants, devi prima creare un'istanza S3 Access Grants uguale ai tuoi dati S3. Regione AWS

Prerequisiti

Se desideri concedere l'accesso ai tuoi dati S3 utilizzando le identità della tua directory aziendale, aggiungi la tua directory di identità aziendale a AWS IAM Identity Center. Se ritieni che non sia ancora il momento per farlo, puoi associare la tua istanza S3 Access Grants a un'istanza del Centro identità IAM in un secondo momento.

Per creare un'istanza S3 Access Grants

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nella barra di navigazione, scegli il nome del file attualmente visualizzato Regione AWS. Quindi, scegli la regione a cui vuoi passare.
3. Nel pannello di navigazione a sinistra, scegli Access Grants.
4. Nella pagina S3 Access Grants, scegli Crea un'istanza S3 Access Grants.

    a. Nel Passaggio 1 della procedura guidata Configura l'istanza Access Grants, verifica di voler creare l'istanza nella Regione AWS corrente. Assicurati che sia la stessa Regione AWS in cui si trovano i tuoi dati S3. Puoi creare un'istanza S3 Access Grants per account. Regione AWS

    b. (Facoltativo) Se hai aggiunto la tua directory di identità aziendale a AWS IAM Identity Center, puoi associare questa istanza IAM Identity Center della tua directory aziendale alla tua istanza S3 Access Grants.

    Per farlo, seleziona Aggiungi istanza del Centro identità IAM in ***regione***. Quindi inserisci il nome della risorsa Amazon (ARN) dell'istanza del Centro identità IAM.

    Se non hai ancora aggiunto la tua directory di identità aziendale al Centro identità IAM, puoi associare l'istanza S3 Access Grants a un'istanza del Centro identità IAM in un secondo momento.

    c. Per creare l'istanza S3 Access Grants, scegli Avanti. Per registrare una posizione, consulta Passaggio 2: registrare una posizione.
5. Se l'opzione Avanti o Crea istanza S3 Access Grants è disabilitata:

Non puoi creare un'istanza

- Potresti avere già un'istanza S3 Access Grants nella stessa Regione AWS. Nel pannello di navigazione a sinistra, scegli Access Grants. Nella pagina S3 Access Grants, scorri verso il basso fino alla sezione Istanza S3 Access Grants nel tuo account per stabilire se un'istanza esiste già.
- Potresti non disporre dell'autorizzazione `s3:CreateAccessGrantsInstance` richiesta per creare un'istanza S3 Access Grants. Contatta l'amministratore dell'account. Per le autorizzazioni aggiuntive necessarie per associare un'istanza del Centro identità IAM alla tua istanza S3 Access Grants, consulta CreateAccessGrantsInstance.

Utilizzando il AWS CLI

Per installare AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Per utilizzare il seguente comando di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

Example Creazione di un'istanza S3 Access Grants

```
aws s3control create-access-grants-instance \
--account-id 111122223333 \
--region us-east-2
```

Risposta:

```
{
    "CreatedAt": "2023-05-31T17:54:07.893000+00:00",
    "AccessGrantsInstanceId": "default",
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default"
}
```

Utilizzo di REST API

Puoi utilizzare la REST API di Amazon S3 per creare un'istanza S3 Access Grants. Per informazioni sul supporto REST API per la gestione di un'istanza S3 Access Grants, consulta le sezioni seguenti nella Documentazione di riferimento delle API di Amazon Simple Storage Service:

- AssociateAccessGrantsIdentityCenter
- CreateAccessGrantsInstance
- DeleteAccessGrantsInstance
- DissociateAccessGrantsIdentityCenter
- GetAccessGrantsInstance
- GetAccessGrantsInstanceForPrefix
- GetAccessGrantsInstanceResourcePolicy
- ListAccessGrantsInstances
- PutAccessGrantsInstanceResourcePolicy

Utilizzo degli AWS SDK

Questa sezione fornisce un esempio di come creare un'istanza S3 Access Grants utilizzando gli AWS SDK.

Java

Questo esempio crea l'istanza S3 Access Grants, che funge da container per le tue concessioni di accesso individuali. Puoi avere un'istanza S3 Access Grants per account Regione AWS . La risposta include l'ID dell'istanza `default` e un nome della risorsa Amazon (ARN) generato per la tua istanza S3 Access Grants.

Example Creazione di una richiesta di istanza S3 Access Grants

```
public void createAccessGrantsInstance() {
CreateAccessGrantsInstanceRequest createRequest =
 CreateAccessGrantsInstanceRequest.builder().accountId("111122223333").build();
CreateAccessGrantsInstanceResponse createResponse =
 s3Control.createAccessGrantsInstance(createRequest);LOGGER.info("CreateAccessGrantsInstance
 " + createResponse);
}
```

Risposta:

```
CreateAccessGrantsInstanceResponse(
CreatedAt=2023-06-07T01:46:20.507Z,
AccessGrantsInstanceId=default,
```

```
AccessGrantsInstanceArn=arn:aws:s3:us-east-2:111122223333:access-grants/default)
```

Argomenti

- Visualizza i dettagli di un'istanza S3 Access Grants
- Associazione o annullamento dell'associazione dell'istanza del Centro identità IAM
- Eliminazione di un'istanza S3 Access Grants

Visualizza i dettagli di un'istanza S3 Access Grants

Puoi visualizzare i dettagli della tua istanza Amazon S3 Access Grants in una particolare Regione AWS. Puoi anche elencare le tue istanze S3 Access Grants, incluse le istanze che sono state condivise con te tramite (). AWS Resource Access Manager AWS RAM

Puoi visualizzare i dettagli della tua istanza S3 Access Grants o elencare le tue istanze S3 Access Grants utilizzando la console Amazon S3, la () AWS Command Line Interface ,AWS CLI l'API REST di Amazon S3 e gli SDK. AWS

Utilizzo della console S3

Per visualizzare un'istanza S3 Access Grants

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Access Grants.
3. Nella pagina S3 Access Grants, scegli la regione che contiene l'istanza S3 Access Grants con cui vuoi lavorare.
4. La pagina S3 Access Grants elenca le istanze S3 Access Grants e tutte le istanze multi-account che sono state condivise con il tuo account. Per visualizzare i dettagli di un'istanza, scegli Visualizza dettagli.

Usando il AWS CLI

Per installare AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Per utilizzare il seguente comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

Example – Ottieni i dettagli di un'istanza S3 Access Grants

```
aws s3control get-access-grants-instance \
 --account-id 111122223333 \
 --region us-east-2
```

Risposta:

```
{
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2: 111122223333:access-grants/
default",
    "AccessGrantsInstanceId": "default",
    "CreatedAt": "2023-05-31T17:54:07.893000+00:00"
}
```

Example – Elenca tutte le istanze S3 Access Grants relative a un account

Questa azione elenca le istanze S3 Access Grants per un account. Puoi avere solo un'istanza S3 Access Grants per. Regione AWS Questa azione elenca anche altre istanze S3 Access Grants multi-account a cui il tuo account ha accesso.

```
aws s3control list-access-grants-instances \
 --account-id 111122223333 \
 --region us-east-2
```

Risposta:

```
{
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2: 111122223333:access-grants/
default",
    "AccessGrantsInstanceId": "default",
    "CreatedAt": "2023-05-31T17:54:07.893000+00:00"
}
```

Utilizzo di REST API

Per informazioni sul supporto REST API Amazon S3 per la gestione di un'istanza S3 Access Grants, consulta le sezioni seguenti nella Documentazione di riferimento delle API di Amazon Simple Storage Service:

- GetAccessGrantsInstance

- GetAccessGrantsInstanceForPrefix
- ListAccessGrantsInstances

Utilizzo degli SDK AWS

Questa sezione fornisce esempi su come ottenere i dettagli di un'istanza di S3 Access Grants utilizzando gli SDK. AWS

Per utilizzare i seguenti esempi, sostituisci *user input placeholders* con le tue informazioni.

Java

Example – Ottieni un'istanza S3 Access Grants

```
public void getAccessGrantsInstance() {
GetAccessGrantsInstanceRequest getRequest = GetAccessGrantsInstanceRequest.builder()
.accountId("111122223333")
.build();
GetAccessGrantsInstanceResponse getResponse =
 s3Control.getAccessGrantsInstance(getRequest);
LOGGER.info("GetAccessGrantsInstanceResponse: " + getResponse);
}
```

Risposta:

```
GetAccessGrantsInstanceResponse(
AccessGrantsInstanceArn=arn:aws:s3:us-east-2: 111122223333:access-grants/default,
CreatedAt=2023-06-07T01:46:20.507Z)
```

Example – Elenca tutte le istanze S3 Access Grants relative a un account

Questa azione elenca le istanze S3 Access Grants per un account. Puoi avere una sola istanza S3 Access Grants per regione. Questa azione può elencare anche altre istanze S3 Access Grants multi-account a cui il tuo account ha accesso.

```
public void listAccessGrantsInstances() {
ListAccessGrantsInstancesRequest listRequest =
 ListAccessGrantsInstancesRequest.builder()
.accountId("111122223333")
.build();
```

```
ListAccessGrantsInstancesResponse listResponse =
 s3Control.listAccessGrantsInstances(listRequest);
LOGGER.info("ListAccessGrantsInstancesResponse: " + listResponse);
}
```

Risposta:

```
ListAccessGrantsInstancesResponse(
AccessGrantsInstancesList=[
ListAccessGrantsInstanceEntry(
AccessGrantsInstanceId=default,
AccessGrantsInstanceArn=arn:aws:s3:us-east-2:111122223333:access-grants/default,
CreatedAt=2023-06-07T04:28:11.728Z
)
]
)
```

Associazione o annullamento dell'associazione dell'istanza del Centro identità IAM

In Amazon S3 Access Grants, puoi associare l' AWS IAM Identity Center istanza della tua directory di identità aziendale a un'istanza S3 Access Grants. Dopo averlo fatto, puoi creare concessioni di accesso per gli utenti e i gruppi della tua directory aziendale, oltre agli utenti e ai ruoli AWS Identity and Access Management (IAM).

Se non desideri più creare concessioni di accesso per gli utenti e i gruppi della tua directory aziendale, puoi annullare l'associazione dell'istanza del Centro identità IAM dall'istanza S3 Access Grants.

Puoi associare o annullare l'associazione di un'istanza del Centro identità IAM utilizzando la console Amazon S3, l' AWS Command Line Interface (AWS CLI), la REST API di Amazon S3 e gli AWS SDK.

Utilizzo della console S3

Prima di associare l'istanza del Centro identità IAM all'istanza S3 Access Grants, devi aggiungere la directory di identità aziendale al Centro identità IAM. Per ulteriori informazioni, consulta the section called “S3 Access Grants e identità delle directory aziendali”.

Per associare un'istanza S3 Access Grants a un'istanza del Centro identità IAM

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

2. Nel pannello di navigazione a sinistra, scegli Access Grants.
3. Nella pagina S3 Access Grants, scegli la regione che contiene l'istanza S3 Access Grants con cui vuoi lavorare.
4. Scegli Visualizza dettagli per l'istanza.
5. Nella pagina dei dettagli, nella sezione Centro identità IAM, scegli di aggiungere un'istanza del Centro identità IAM o di annullare la registrazione di un'istanza del Centro identità IAM già associata.

Usando il AWS CLI

Per installare AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Per utilizzare il seguente comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

Example – Associa un'istanza S3 Access Grants a un'istanza del Centro identità IAM

```
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 – Annulla associazione di un'istanza S3 Access Grants da un'istanza del Centro identità IAM

```
aws s3control dissociate-access-grants-identity-center \
 --account-id 111122223333 \
 --profile access-grants-profile \
 --region eu-central-1

// No response body
```

Utilizzo di REST API

Per informazioni sul supporto REST API di Amazon S3 per la gestione dell'associazione tra un'istanza del Centro identità IAM e un'istanza S3 Access Grants, consulta le sezioni seguenti nella Documentazione di riferimento delle API di Amazon Simple Storage Service:

- AssociateAccessGrantsIdentityCenter
- DissociateAccessGrantsIdentityCenter

Eliminazione di un'istanza S3 Access Grants

Puoi eliminare un'istanza Amazon S3 Access Grants da un account Regione AWS . Tuttavia, prima di eliminare un'istanza S3 Access Grants, devi eseguire queste operazioni:

- Elimina tutte le risorse all'interno dell'istanza S3 Access Grants, incluse tutte le concessioni e le posizioni. Per ulteriori informazioni, consulta Eliminazione di una concessione ed Eliminazione di una posizione.
- Se hai associato un' AWS IAM Identity Center istanza alla tua istanza S3 Access Grants, devi dissociare l'istanza IAM Identity Center. Per ulteriori informazioni, consulta Associazione o annullamento dell'associazione dell'istanza del Centro identità IAM.

> **Important**
>
> Se elimini un'istanza S3 Access Grants, l'eliminazione è permanente e non può essere annullata. Tutti gli assegnatari a cui è stato fornito l'accesso grazie alle concessioni in questa istanza S3 Access Grants perderanno l'accesso ai tuoi dati S3.

Puoi eliminare un'istanza S3 Access Grants utilizzando la console Amazon S3,AWS CLI(), AWS Command Line Interface l'API REST di Amazon S3 e gli SDK. AWS

Utilizzo della console S3

Per eliminare un'istanza S3 Access Grants

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Access Grants.
3. Nella pagina S3 Access Grants, scegli la regione che contiene l'istanza S3 Access Grants con cui vuoi lavorare.
4. Scegli Visualizza dettagli per l'istanza.
5. Nella pagina dei dettagli dell'istanza scegli Elimina istanza nell'angolo in alto a destra.

6. Nella finestra di dialogo visualizzata, seleziona Elimina. Questa operazione non può essere annullata.

Usando il AWS CLI

Per installare AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Per utilizzare il seguente comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

> **Note**
>
> Prima di poter eliminare un'istanza S3 Access Grants, devi eliminare tutte le concessioni e le posizioni create all'interno dell'istanza S3 Access Grants. Se hai associato un'istanza del Centro identità IAM alla tua istanza S3 Access Grants, devi prima annullare l'associazione dell'istanza del Centro identità IAM.

Example – Elimina un'istanza 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
```

Utilizzo di REST API

Per informazioni sul supporto REST API di Amazon S3 per l'eliminazione di un'istanza S3 Access Grants, consulta DeleteAccessGrantsInstance nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Utilizzo degli AWS SDK

Questa sezione fornisce esempi di come eliminare un'istanza S3 Access Grants utilizzando gli AWS SDK.

Per utilizzare il seguente esempio, sostituisci *user input placeholders* con le tue informazioni.

Java

> **Note**
>
> Prima di poter eliminare un'istanza S3 Access Grants, devi eliminare tutte le concessioni e le posizioni create all'interno dell'istanza S3 Access Grants. Se hai associato un'istanza del Centro identità IAM alla tua istanza S3 Access Grants, devi prima annullare l'associazione dell'istanza del Centro identità IAM.

Example – Elimina un'istanza S3 Access Grants

```
public void deleteAccessGrantsInstance() {
DeleteAccessGrantsInstanceRequest deleteRequest =
 DeleteAccessGrantsInstanceRequest.builder()
.accountId("111122223333")
.build();
DeleteAccessGrantsInstanceResponse deleteResponse =
 s3Control.deleteAccessGrantsInstance(deleteRequest);
LOGGER.info("DeleteAccessGrantsInstanceResponse: " + deleteResponse);
}
```

Registrazione di una posizione

Dopo aver creato un'istanza Amazon S3 Access Grants Regione AWS nel tuo account, puoi registrare una posizione S3 in quell'istanza. Una posizione è una risorsa S3 che contiene dati a cui desideri concedere l'accesso. Puoi registrare la posizione predefinita`s3://`, ovvero tutti i bucket in cui risiedi, e restringere l' Regione AWS ambito di accesso in un secondo momento, quando crei concessioni di accesso individuali. Puoi anche registrare un bucket specifico o un bucket e prefisso come posizione.

Devi prima registrare almeno una posizione con la tua istanza S3 Access Grants prima di poter creare concessioni di accesso. Quando registri una posizione, devi anche specificare il ruolo AWS Identity and Access Management (IAM) che S3 Access Grants assume per soddisfare le richieste di runtime per la posizione e definire l'ambito delle autorizzazioni fino alla concessione specifica al runtime.

URI S3	Ruolo IAM	Descrizione
`s3://`	*`Default-IAM-role`*	La posizione predefinita, `s3://`, include tutti i bucket nella Regione AWS.
`s3://`*`example-s3-bucket1`*`/`	*`IAM-role-For-bucket`*	Questa posizione include tutti gli oggetti nel bucket specificato.

Prima di poter registrare una posizione, assicurati di eseguire la seguente procedura:

- Crea uno o più bucket contenenti i dati a cui desideri concedere l'accesso. Questi bucket devono trovarsi nella stessa istanza di S3 Regione AWS Access Grants. Per ulteriori informazioni, consulta Creazione di un bucket.

  Per aggiungere un prefisso a un bucket, consulta Creazione dei nomi delle chiavi degli oggetti.

- Crea un ruolo IAM e fornisci al principale del servizio S3 Access Grants l'accesso a questo ruolo nel file di policy delle risorse. A questo proposito, puoi creare un file JSON contenente le istruzioni elencate di seguito. Per aggiungere la policy della risorsa al tuo account, consulta Creazione e collegamento della prima policy gestita dal cliente.

  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"}
      }
    ]
  }
  ```

- Creare una policy IAM per collegare le autorizzazioni di Amazon S3 al ruolo IAM. Consulta il seguente file `iam-policy.json` di esempio e sostituisci *`user input placeholders`* con le tue informazioni.

Note

- Se utilizzi la crittografia lato server con chiavi AWS Key Management Service (AWS KMS) per crittografare i tuoi dati, l'esempio seguente include le AWS KMS autorizzazioni necessarie per il ruolo IAM nella policy. Se non utilizzi questa funzionalità, puoi rimuovere queste autorizzazioni dalla tua policy IAM.
- Puoi limitare il ruolo IAM all'accesso ai dati S3 solo se le credenziali vengono fornite da S3 Access Grants. Questo esempio mostra come aggiungere un'`Condition`istruzione per una specifica istanza di S3 Access Grants. Per fare ciò, sostituisci l'ARN dell'istanza S3 Access Grants nella dichiarazione delle condizioni con l'ARN dell'istanza S3 Access Grants, che ha il 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:RegioneAWS: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:RegioneAWS:accountId:access-grants/default"]
            }
         }
      },
      {
         "Sid": "KMSPermissions",
         "Effect":"Allow",
         "Action":[
```

```
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Puoi registrare una posizione nella tua istanza S3 Access Grants utilizzando la console Amazon S3,AWS CLI(), AWS Command Line Interface l'API REST di Amazon S3 o gli SDK. AWS

Utilizzo della console S3

Prima di poter concedere l'accesso ai dati S3 con S3 Access Grants, devi avere almeno una posizione registrata.

Per registrare una posizione nella tua istanza S3 Access Grants

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Access Grants.
3. Nella pagina S3 Access Grants, scegli la regione che contiene l'istanza S3 Access Grants con cui vuoi lavorare.

   Se utilizzi un'istanza S3 Access Grants per la prima volta, assicurati di aver completato il Passaggio 1: crea un'istanza S3 Access Grants e di aver eseguito il Passaggio 2 della procedura guidata Configurazione dell'istanza Access Grants. Se disponi già di un'istanza S3 Access Grants, seleziona Visualizza dettagli, quindi dalla scheda Posizioni, seleziona Registra posizione.

   a. Per Ambito della posizione, seleziona Sfoglia S3 o inserisci il percorso URI S3 della posizione che desideri registrare. Per i formati URI S3, consulta la tabella dei formati di posizione. Dopo aver inserito un URI, puoi scegliere Visualizza per andare alla posizione.
   b. In Ruolo IAM, scegliere una delle seguenti opzioni:

      - Scegli tra i ruoli IAM esistenti

        Scegli un ruolo IAM dall'elenco a discesa. Dopo aver scelto un ruolo, scegli Visualizza per avere la certezza che questo ruolo disponga delle autorizzazioni necessarie per gestire

la posizione che stai registrando. In particolare, assicurati che questo ruolo conceda a S3 Access Grants le autorizzazioni `sts:AssumeRole` e `sts:SetSourceIdentity`.

- Inserisci l'ARN del ruolo IAM

  Accedi alla console IAM. Copia l'Amazon Resource Name (ARN) del ruolo IAM e incollalo in questa casella.

c. Per finire, scegli Avanti o Registra posizione.

4. Risoluzione dei problemi

   Impossibile registrare la posizione

   - La posizione potrebbe essere già registrata.

     Potresti non avere l'autorizzazione `s3:CreateAccessGrantsLocation` per registrare le posizioni. Contatta l'amministratore dell'account.

Usando il AWS CLI

Per installare AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Puoi registrare la posizione predefinita, `s3://`, o una posizione personalizzata nella tua istanza S3 Access Grants. Assicurati di creare prima un ruolo IAM con accesso del principale alla posizione, quindi assicurati di concedere a S3 Access Grants l'autorizzazione ad assumere questo ruolo.

Per utilizzare i seguenti comandi di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

Example Creazione di una policy di risorse

Crea una policy che consenta a S3 Access Grants di assumere il ruolo IAM. A questo proposito, puoi creare un file JSON contenente le istruzioni elencate di seguito. Per aggiungere la policy della risorsa al tuo account, consulta Creazione e collegamento della prima policy gestita dal 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 Creazione del ruolo

Per creare il ruolo, esegui il comando IAM seguente.

```
aws iam create-role --role-name accessGrantsTestRole \
 --region us-east-2 \
 --assume-role-policy-document file://TestRolePolicy.json
```

L'esecuzione del comando `create-role` restituisce la policy:

```
{
    "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

Creare una policy IAM per collegare le autorizzazioni di Amazon S3 al ruolo IAM. Consulta il seguente file `iam-policy.json` di esempio e sostituisci *`user input placeholders`* con le tue informazioni.

> **Note**
>
> Se utilizzi la crittografia lato server con chiavi AWS Key Management Service (AWS KMS) per crittografare i tuoi dati, l'esempio seguente aggiunge le AWS KMS autorizzazioni necessarie per il ruolo IAM nella policy. Se non utilizzi questa funzionalità, puoi rimuovere queste autorizzazioni dalla tua policy IAM.
> Per avere la certezza che il ruolo IAM possa essere usato per accedere ai dati in S3 solo se le credenziali sono distribuite da S3 Access Grants, questo esempio mostra come aggiungere un'istruzione `Condition` che specifichi l'istanza S3 Access Grants (`s3:AccessGrantsInstance:` *`InstanceArn`*) nella policy IAM. Quando utilizzi la seguente policy di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

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:Regione
 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:Regione
 AWS:accountId:access-grants/default"]
            }
         }
```

```
      },
      {
         "Sid": "KMSPermissions",
         "Effect":"Allow",
         "Action":[
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Example

Esegui il comando seguente:

```
aws iam put-role-policy \
--role-name accessGrantsTestRole \
--policy-name accessGrantsTestRole \
--policy-document file://iam-policy.json
```

Example Registra la posizione predefinita

```
aws s3control create-access-grants-location \
 --account-id 111122223333 \
 --location-scope s3:// \
 --iam-role-arn arn:aws:iam::111122223333:role/accessGrantsTestRole
```

Risposta:

```
{"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 Registra una posizione personalizzata

```
aws s3control create-access-grants-location \
 --account-id 111122223333 \
 --location-scope s3://DOC-BUCKET-EXAMPLE/ \
 --iam-role-arn arn:aws:iam::123456789012:role/accessGrantsTestRole
```

Risposta:

```
{"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"
}
```

Utilizzo di REST API

Per informazioni sul supporto REST API di Amazon S3 per la gestione di un'istanza S3 Access Grants, consulta le sezioni seguenti nella Documentazione di riferimento delle API di Amazon Simple Storage Service:

- CreateAccessGrantsLocation
- DeleteAccessGrantsLocation
- GetAccessGrantsLocation
- ListAccessGrantsLocations
- UpdateAccessGrantsLocation

Utilizzo degli SDK AWS

Questa sezione fornisce esempi di come registrare posizioni tramite gli AWS SDK.

Per utilizzare i seguenti esempi, sostituisci *`user input placeholders`* con le tue informazioni.

Java

Puoi registrare la posizione predefinita, `s3://`, o una posizione personalizzata nella tua istanza S3 Access Grants. Assicurati di creare prima un ruolo IAM con accesso del principale alla

posizione, quindi assicurati di concedere a S3 Access Grants l'autorizzazione ad assumere questo ruolo.

Per utilizzare i seguenti comandi di esempio, sostituisci *user input placeholders* con le tue informazioni.

Example Registra una posizione predefinita

Richiesta:

```
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);
}
```

Risposta:

```
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 Registra una posizione personalizzata

Richiesta:

```
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);
}
```

Risposta:

```
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
)
```

Argomenti

- Visualizza i dettagli di una posizione registrata
- Aggiornamento di una posizione registrata
- Eliminazione di una posizione registrata

Visualizza i dettagli di una posizione registrata

Puoi ottenere i dettagli di una posizione registrata nella tua istanza S3 Access Grants utilizzando la console Amazon S3, il AWS CLI(), AWS Command Line Interface l'API REST di Amazon S3 e gli SDK. AWS

Utilizzo della console S3

Per visualizzare le posizioni registrate nell'istanza S3 Access Grants

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Access Grants.
3. Nella pagina S3 Access Grants, scegli la regione che contiene l'istanza S3 Access Grants con cui vuoi lavorare.
4. Scegli Visualizza dettagli per l'istanza.

5. Nella pagina dei dettagli dell'istanza, scegli la scheda Posizioni.
6. Trova la posizione registrata che desideri visualizzare. Per filtrare l'elenco delle posizioni registrate, usa la casella di ricerca.

Usando il AWS CLI

Per installare AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Per utilizzare il seguente comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

Example – Ottieni i dettagli di una posizione registrata

```
aws s3control get-access-grants-location \
--account-id 111122223333 \
--access-grants-location-id default
```

Risposta:

```
{
    "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 – Elenca tutte le posizioni registrate in un'istanza di S3 Access Grants

Per limitare i risultati a un prefisso o un bucket S3, puoi opzionalmente utilizzare il parametro `--location-scope s3://bucket-and-or-prefix`.

```
aws s3control list-access-grants-locations \
--account-id 111122223333 \
--region us-east-2
```

Risposta:

```
{"AccessGrantsLocationsList": [
```

```
  {
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "default",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/location/default",
    "LocationScope": "s3://"
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
     },
  {
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb456",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/location/635f1139-1af2-4e43-8131-a4de006eb888",
    "LocationScope": "s3://DOC-EXAMPLE-BUCKET/prefixA*",
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
     }
   ]
  }
```

Utilizzo di REST API

Per informazioni sul supporto REST API di Amazon S3 per ottenere i dettagli di una posizione registrata o elencare tutte le posizioni registrate con un'istanza S3 Access Grants, consulta le seguenti sezioni nella Documentazione di riferimento delle API di Amazon Simple Storage Service:

- GetAccessGrantsLocation
- ListAccessGrantsLocations

Utilizzo degli AWS SDK

Questa sezione fornisce esempi di come ottenere i dettagli di una posizione registrata o elencare tutte le posizioni registrate in un'istanza S3 Access Grants utilizzando gli AWS SDK.

Per utilizzare i seguenti esempi, sostituisci *`user input placeholders`* con le tue informazioni.

Java

Example – Ottieni i dettagli di una posizione registrata

```
public void getAccessGrantsLocation() {
GetAccessGrantsLocationRequest getAccessGrantsLocationRequest =
 GetAccessGrantsLocationRequest.builder()
```

```
.accountId("111122223333")
.accessGrantsLocationId("default")
.build();
GetAccessGrantsLocationResponse getAccessGrantsLocationResponse =
 s3Control.getAccessGrantsLocation(getAccessGrantsLocationRequest);
LOGGER.info("GetAccessGrantsLocationResponse: " + getAccessGrantsLocationResponse);
}
```

Risposta:

```
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 – Elenca tutte le posizioni registrate in un'istanza S3 Access Grants

Per limitare i risultati a un prefisso o un bucket S3, puoi opzionalmente passare un URI S3, ad esempio `s3://bucket-and-or-prefix`, nel parametro `LocationScope`.

```
public void listAccessGrantsLocations() {

ListAccessGrantsLocationsRequest listRequest =
 ListAccessGrantsLocationsRequest.builder()
.accountId("111122223333")
.build();

ListAccessGrantsLocationsResponse listResponse =
 s3Control.listAccessGrantsLocations(listRequest);
LOGGER.info("ListAccessGrantsLocationsResponse: " + listResponse);
}
```

Risposta:

```
ListAccessGrantsLocationsResponse(
AccessGrantsLocationsList=[
ListAccessGrantsLocationsEntry(
CreatedAt=2023-06-07T04:35:11.027Z,
```

```
AccessGrantsLocationId=default,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/default,
LocationScope=s3://,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
),
ListAccessGrantsLocationsEntry(
CreatedAt=2023-06-07T04:35:10.027Z,
AccessGrantsLocationId=635f1139-1af2-4e43-8131-a4de006eb456,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/635f1139-1af2-4e43-8131-a4de006eb888,
LocationScope=s3://DOC-EXAMPLE-BUCKET/prefixA*,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
]
)
```

Aggiornamento di una posizione registrata

Puoi aggiornare il ruolo AWS Identity and Access Management (IAM) di una location registrata nella tua istanza Amazon S3 Access Grants. Per ogni nuovo ruolo IAM che utilizzi per registrare una posizione in S3 Access Grants, assicurati di consentire al principale (`access-grants.s3.amazonaws.com`) del servizio di S3 Access Grants l'accesso a questo ruolo. Per fare ciò, aggiungi una voce per il nuovo ruolo IAM nello stesso file JSON della policy di attendibilità che hai usato quando hai registrato la posizione per la prima volta.

Puoi aggiornare una posizione nella tua istanza S3 Access Grants utilizzando la console Amazon S3,AWS CLI(), AWS Command Line Interface l'API REST di Amazon S3 e gli SDK. AWS

Utilizzo della console S3

Per aggiornare il ruolo IAM di una posizione registrata con la tua istanza S3 Access Grants

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Access Grants.
3. Nella pagina S3 Access Grants, scegli la regione che contiene l'istanza S3 Access Grants con cui vuoi lavorare.
4. Scegli Visualizza dettagli per l'istanza.
5. Nella pagina dei dettagli dell'istanza, scegli la scheda Posizioni.

6. Trova la posizione che intendi aggiornare. Per filtrare l'elenco delle posizioni, usa la casella di ricerca.
7. Scegli il pulsante delle opzioni accanto alla posizione registrata che desideri aggiornare.
8. Aggiorna il ruolo IAM, quindi scegli Salva modifiche.

Usando il AWS CLI

Per installare AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Per utilizzare il seguente comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

Example – Aggiorna il ruolo IAM di una posizione registrata

```
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
```

Risposta:

```
{
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb999",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:777788889999:access-grants/
default/location/635f1139-1af2-4e43-8131-a4de006eb888",
    "LocationScope": "s3://DOC-EXAMPLE-BUCKET/prefixB*",
    "IAMRoleArn": "arn:aws:iam::777788889999:role/accessGrantsTestRole"
}
```

Utilizzo di REST API

Per informazioni sul supporto REST API di Amazon S3 per l'aggiornamento della posizione in un'istanza S3 Access Grants, consulta UpdateAccessGrantsLocation nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Utilizzo degli AWS SDK

Questa sezione fornisce esempi di come aggiornare il ruolo IAM di una sede registrata utilizzando gli AWS SDK.

Per utilizzare il seguente esempio, sostituisci *user input placeholders* con le tue informazioni.

Java

Example – Aggiorna il ruolo IAM di una posizione registrata

```
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);
}
```

Risposta:

```
UpdateAccessGrantsLocationResponse(
CreatedAt=2023-06-07T04:35:10.027Z,
AccessGrantsLocationId=635f1139-1af2-4e43-8131-a4de006eb999,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:777788889999:access-grants/default/
location/635f1139-1af2-4e43-8131-a4de006eb888,
LocationScope=s3://DOC-EXAMPLE-BUCKET/prefixB*,
IAMRoleArn=arn:aws:iam::777788889999:role/accessGrantsTestRole
)
```

Eliminazione di una posizione registrata

Puoi eliminare la registrazione di una posizione da un'istanza Amazon S3 Access Grants. L'eliminazione della posizione ne annulla la registrazione dall'istanza S3 Access Grants.

Prima di poter rimuovere una registrazione di una posizione da un'istanza S3 Access Grants, devi eliminare tutte le concessioni associate a questa posizione. Per informazioni sull'eliminazione delle concessioni, consulta Elimina una concessione.

Puoi eliminare una posizione nella tua istanza S3 Access Grants utilizzando la console Amazon S3,AWS CLI(), AWS Command Line Interface l'API REST di Amazon S3 e gli SDK. AWS

Utilizzo della console S3

Per eliminare la registrazione di una posizione dalla tua istanza Amazon S3 Access Grants

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Access Grants.
3. Nella pagina S3 Access Grants, scegli la regione che contiene l'istanza S3 Access Grants con cui vuoi lavorare.
4. Scegli Visualizza dettagli per l'istanza.
5. Nella pagina dei dettagli dell'istanza, scegli la scheda Posizioni.
6. Trova la posizione che intendi aggiornare. Per filtrare l'elenco delle posizioni, usa la casella di ricerca.
7. Scegli il pulsante di opzione accanto alla posizione registrata che desideri eliminare.
8. Scegli Annulla registrazione.
9. Viene visualizzata una finestra di dialogo che avverte che questa azione non può essere annullata. Per eliminare la posizione, scegli Annulla registrazione.

Usando il AWS CLI

Per installare AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Per utilizzare il seguente comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

Example – Elimina la registrazione di una posizione

```
aws s3control delete-access-grants-location \
--account-id 111122223333 \
--access-grants-location-id  a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
 // No response body
```

Utilizzo di REST API

Per informazioni sul supporto REST API di Amazon S3 per l'eliminazione di una posizione da un'istanza S3 Access Grants, consulta DeleteAccessGrantsLocation nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Utilizzo degli AWS SDK

Questa sezione fornisce esempi su come eliminare una posizione tramite gli AWS SDK.

Per utilizzare il seguente esempio, sostituisci *user input placeholders* con le tue informazioni.

Java

Example – Elimina la registrazione di una posizione

```
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);
}
```

Risposta:

```
DeleteAccessGrantsLocationResponse()
```

Creazione di concessioni

Una volta che hai registrato almeno una posizione nella tua istanza S3 Access Grants, puoi creare una concessione di accesso. Una concessione di accesso fornisce all'assegnatario l'autorizzazione ad accedere a una posizione registrata.

Il beneficiario può essere un utente o un ruolo AWS Identity and Access Management (IAM) oppure un utente o un gruppo di directory. Un utente di directory è un utente della directory aziendale o di una origine di identità esterna che hai aggiunto all'istanza AWS IAM Identity Center che è associata all'istanza S3 Access Grants. Per creare una concessione per un utente o un gruppo specifico dal Centro identità IAM, trova il GUID utilizzato dal Centro identità IAM per identificare quell'utente nel Centro identità IAM, ad esempio `a1b2c3d4-5678-90ab-cdef-EXAMPLE11111`.

Puoi concedere l'accesso a un bucket, a un prefisso o a un oggetto. Un prefisso in Amazon S3 è una stringa di caratteri all'inizio del nome della chiave di un oggetto che viene utilizzata per organizzare

gli oggetti all'interno di un bucket. Può trattarsi di qualsiasi stringa di caratteri consentiti, ad esempio i nomi delle chiavi degli oggetti in un bucket che iniziano con il prefisso `engineering/`.

Sottoprefisso

Quando concedi l'accesso a una posizione registrata, puoi utilizzare il campo `Subprefix` per restringere l'ambito a un prefisso specifico all'interno di un bucket o a un oggetto specifico in un bucket.

Non puoi creare una concessione di accesso per la posizione `s3://` predefinita, che consentirebbe all'assegnatario di accedere a tutti i bucket di una regione. Se scegli la posizione `s3://` predefinita come posizione della concessione, devi restringere l'ambito della concessione utilizzando il campo `Subprefix` per specificare una delle seguenti opzioni:

- Un bucket: `s3://`*`bucket`*`/*`
- Un prefisso all'interno di un bucket: `s3://`*`bucket`*`/`*`prefix`*`*`
- Un prefisso all'interno di un prefisso: `s3://`*`bucket`*`/`*`prefixA`*`/`*`prefixB`*`*`
- Un oggetto: `s3://`*`bucket`*`/`*`object-key-name`*.

Se crei una concessione di accesso in cui la posizione registrata è un bucket, puoi inserire uno dei seguenti valori nel campo `Subprefix`:

- Un prefisso all'interno di un bucket: *`prefix`*`*`
- Un prefisso all'interno di un prefisso: *`prefixA/prefixB`*`*`
- Un oggetto: `/`*`object-key-name`*.

L'ambito della concessione mostrato nella console Amazon S3 o `GrantScope` quello restituito nella risposta API o AWS Command Line Interface (AWS CLI) è il risultato della concatenazione del percorso di posizione con. `Subprefix` Assicurati che questo percorso concatenato sia mappato correttamente al bucket, al prefisso o all'oggetto S3 a cui desideri concedere l'accesso.

Se stai creando una concessione di accesso per un solo oggetto, specifica nella chiamata API o nel comando CLI che `s3PrefixType` è `Object`.

> **Note**
>
> Non puoi creare una concessione a un bucket se il bucket non esiste ancora. Tuttavia, puoi creare una concessione a un prefisso che non esiste ancora.

Puoi creare una concessione di accesso utilizzando la console Amazon S3 AWS CLI, l'API REST di Amazon S3 e gli SDK. AWS

Utilizzo della console S3

Per creare una concessione di accesso

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Access Grants.
3. Nella pagina S3 Access Grants, scegli la regione che contiene l'istanza S3 Access Grants con cui vuoi lavorare.

   Se utilizzi un'istanza S3 Access Grants per la prima volta, assicurati di aver completato il Passaggio 2: registra una posizione e di aver eseguito il Passaggio 3 della procedura guidata Configurazione dell'istanza Access Grants. Se hai già un'istanza S3 Access Grants, scegli Visualizza dettagli, quindi dalla scheda Concessioni, scegli Crea concessione.

   a. Nella sezione Ambito della concessione, seleziona o inserisci una posizione registrata.

      Se hai selezionato la posizione `s3://` predefinita, utilizza la casella Sottoprefisso per restringere l'ambito della concessione di accesso. Per ulteriori informazioni, consulta Sottoprefisso. Se concedi l'accesso solo a un oggetto, seleziona L'ambito della concessione è un oggetto.

   b. In Autorizzazioni e accesso, seleziona il livello di autorizzazione, ovvero Lettura, Scrittura o entrambi.

      Quindi seleziona Tipo di assegnatario. Se hai aggiunto la tua directory aziendale al Centro identità IAM e hai associato questa istanza del Centro identità IAM all'istanza S3 Access Grants, puoi scegliere Identità della directory dal Centro identità IAM. Se scegli questa opzione, ottieni l'ID dell'utente o del gruppo dal Centro identità IAM e inseriscilo in questa sezione.

Se Tipo di assegnatario è un utente o un ruolo IAM, scegli Principale IAM. In Tipo di principale IAM, scegli Utente o Ruolo. Quindi, in Utente principale IAM, seleziona dall'elenco o inserisci l'ID dell'identità.

c. Per creare la concessione S3 Access Grants, seleziona Avanti o Crea concessione.

4. Se l'opzione Avanti o Crea concessione è disabilitata:

   Impossibile creare una concessione

   - Potrebbe essere necessario registrare prima una posizione nell'istanza S3 Access Grants.
   - Potresti non disporre dell'autorizzazione `s3:CreateAccessGrant` per creare una concessione di accesso. Contatta l'amministratore dell'account.

Usando il AWS CLI

Per installare AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Gli esempi seguenti mostrano come creare una richiesta di concessione di accesso per un principale IAM e come creare una richiesta di concessione di accesso per un utente o un gruppo della directory aziendale.

Per utilizzare i seguenti comandi di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

> **Note**
>
> Se stai creando una concessione di accesso che conceda l'accesso a un solo oggetto, includi il parametro `--s3-prefix-type Object` richiesto.

Example Crea una richiesta di concessione di accesso per un principale 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 Crea una risposta alla concessione di accesso

```
{"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"
}
```

Creazione di una richiesta di autorizzazione di accesso per un utente o un gruppo di utenti della directory

Per creare una richiesta di concessione di accesso per un utente o un gruppo della directory, è necessario innanzitutto ottenere il GUID per l'utente o il gruppo della directory eseguendo uno dei seguenti comandi.

Example Ottieni un GUID per un utente o un gruppo di utenti della directory

Puoi trovare il GUID di un utente IAM Identity Center tramite la console IAM Identity Center o utilizzando gli AWS SDK AWS CLI o. Il comando seguente elenca gli utenti nell'istanza del Centro identità IAM specificata, con i relativi nomi e identificatori.

```
aws identitystore list-users --identity-store-id d-1a2b3c4d1234
```

Questo comando elenca i gruppi nell'istanza Centro identità IAM specificata.

```
aws identitystore list-groups --identity-store-id d-1a2b3c4d1234
```

Example Creazione di una concessione di accesso per un utente o un gruppo di directory

Questo comando è simile alla creazione di una concessione per utenti o ruoli IAM, tranne per il fatto che il tipo di assegnatario è `DIRECTORY_USER` o `DIRECTORY_GROUP` e l'identificatore dell'assegnatario è il GUID per l'utente o il gruppo di directory.

```
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 \
```

Utilizzo di REST API

Per informazioni sul supporto REST API di Amazon S3 per la gestione delle concessioni di accesso, consulta le sezioni seguenti nella Documentazione di riferimento delle API di Amazon Simple Storage Service:

- CreateAccessGrant
- DeleteAccessGrant
- GetAccessGrant
- ListAccessGrants

Utilizzo degli SDK AWS

Questa sezione fornisce esempi di come creare una concessione di accesso utilizzando gli AWS SDK.

Java

Per utilizzare il seguente esempio, sostituisci *`user input placeholders`* con le tue informazioni.

> **Note**
>
> Se crei una concessione di accesso che conceda l'accesso a un solo oggetto, includi il parametro `.s3PrefixType(S3PrefixType.Object)` richiesto.

Example Crea una risposta alla concessione di accesso

```
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 Crea una risposta alla concessione di accesso

```
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
)
```

Argomenti

- Visualizza una concessione
- Eliminazione di una concessione

Visualizza una concessione

Puoi visualizzare i dettagli di una concessione di accesso nella tua istanza Amazon S3 Access Grants utilizzando la console Amazon S3, il AWS Command Line Interface (), l'API REST di Amazon S3 e AWS CLI gli SDK. AWS

Utilizzo della console S3

Per visualizzare i dettagli di una concessione di accesso

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Access Grants.
3. Nella pagina S3 Access Grants, scegli la regione che contiene l'istanza S3 Access Grants con cui vuoi lavorare.
4. Scegli Visualizza dettagli per l'istanza.
5. Nella pagina Dettagli, scegli la scheda Concessioni.
6. Nella sezione Concessioni, trova la concessione di accesso che desideri visualizzare. Puoi utilizzare la casella di ricerca per filtrare l'elenco delle concessioni.

Usando il AWS CLI

Per installare AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Per utilizzare i seguenti comandi di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

Example – Ottieni i dettagli di una concessione di accesso

```
aws s3control get-access-grant \
--account-id 111122223333 \
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Risposta:

```
{
    "CreatedAt": "2023-05-31T18:41:34.663000+00:00",
    "AccessGrantId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
```

```
    "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "Grantee": {
        "GranteeType": "IAM",
        "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"
    },
    "Permission": "READ",
    "AccessGrantsLocationId": "12a6710f-5af8-41f5-b035-0bc795bf1a2b",
    "AccessGrantsLocationConfiguration": {
        "S3SubPrefix": "prefixB*"
    },
    "GrantScope": "s3://DOC-EXAMPLE-BUCKET/"
}
```

Example – Elenca tutte le concessioni di accesso in un'istanza S3 Access Grants

Facoltativamente, puoi utilizzare i seguenti parametri per limitare i risultati a un prefisso S3 o a un'identità AWS Identity and Access Management (IAM):

- Sottoprefisso: `--grant-scope s3://bucket-name/prefix*`
- Identità IAM: `--grantee-type IAM` e `--grantee-identifier arn:aws:iam::123456789000:role/accessGrantsConsumerRole`

```
aws s3control list-access-grants \
--account-id 111122223333
```

Risposta:

```
{
    "AccessGrantsList": [{"CreatedAt": "2023-06-14T17:54:46.542000+00:00",
            "AccessGrantId": "dd8dd089-b224-4d82-95f6-975b4185bbaa",
            "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/dd8dd089-b224-4d82-95f6-975b4185bbaa",
            "Grantee": {
                "GranteeType": "IAM",
                "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"
            },
            "Permission": "READ",
            "AccessGrantsLocationId": "23514a34-ea2e-4ddf-b425-d0d4bfcarda1",
            "GrantScope": "s3://DOC-EXAMPLE-BUCKET/prefixA*"
        },
```

```
        {"CreatedAt": "2023-06-24T17:54:46.542000+00:00",
            "AccessGrantId": "ee8ee089-b224-4d72-85f6-975b4185a1b2",
            "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/ee8ee089-b224-4d72-85f6-975b4185a1b2",
            "Grantee": {
                "GranteeType": "IAM",
                "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-9"
            },
            "Permission": "READ",
            "AccessGrantsLocationId": "12414a34-ea2e-4ddf-b425-d0d4bfcacao0",
            "GrantScope": "s3://DOC-EXAMPLE-BUCKET/prefixB*"
        },

    ]
}
```

Utilizzo di REST API

Puoi utilizzare le operazioni API di Amazon S3 per visualizzare i dettagli di una concessione di accesso ed elencare tutte le concessioni di accesso in un'istanza S3 Access Grants. Per informazioni sul supporto REST API per la gestione delle concessioni di accesso, consulta le sezioni seguenti nella Documentazione di riferimento delle API di Amazon Simple Storage Service:

- GetAccessGrant
- ListAccessGrants

Utilizzo degli SDK AWS

Questa sezione fornisce esempi di come ottenere i dettagli di una concessione di accesso utilizzando gli AWS SDK.

Per utilizzare i seguenti esempi, sostituisci *`user input placeholders`* con le tue informazioni.

Java

Example – Ottieni i dettagli di una concessione di accesso

```
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);
}
```

Risposta:

```
GetAccessGrantResponse(
CreatedAt=2023-06-07T05:20:26.330Z,
AccessGrantId=a1b2c3d4-5678-90ab-cdef-EXAMPLE22222,
AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant-fd3a5086-42f7-4b34-9fad-472e2942c70e,
Grantee=Grantee(
GranteeType=IAM,
GranteeIdentifier=arn:aws:iam::111122223333:user/data-consumer-3
),
Permission=READ,
AccessGrantsLocationId=12a6710f-5af8-41f5-b035-0bc795bf1a2b,
AccessGrantsLocationConfiguration=AccessGrantsLocationConfiguration(
S3SubPrefix=prefixB*
),
GrantScope=s3://DOC-EXAMPLE-BUCKET/
)
```

Example – Elenca tutte le concessioni di accesso in un'istanza S3 Access Grants

Facoltativamente, puoi utilizzare questi parametri per limitare i risultati a un prefisso S3 o un'identità IAM:

- Ambito: `GrantScope=s3://bucket-name/prefix*`
- Assegnatario: `GranteeType=IAM` e `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);
}
```

Risposta:

```
ListAccessGrantsResponse(
AccessGrantsList=[
ListAccessGrantEntry(
CreatedAt=2023-06-14T17:54:46.540z,
AccessGrantId=dd8dd089-b224-4d82-95f6-975b4185bbaa,
AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/dd8dd089-b224-4d82-95f6-975b4185bbaa,
Grantee=Grantee(
GranteeType=IAM, GranteeIdentifier= arn:aws:iam::111122223333:user/data-consumer-3
),
Permission=READ,
AccessGrantsLocationId=23514a34-ea2e-4ddf-b425-d0d4bfcarda1,
GrantScope=s3://DOC-EXAMPLE-BUCKET/prefixA
),
ListAccessGrantEntry(
CreatedAt=2023-06-24T17:54:46.540Z,
AccessGrantId=ee8ee089-b224-4d72-85f6-975b4185a1b2,
AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/ee8ee089-b224-4d72-85f6-975b4185a1b2,
Grantee=Grantee(
GranteeType=IAM, GranteeIdentifier= arn:aws:iam::111122223333:user/data-consumer-9
),
Permission=READ,
AccessGrantsLocationId=12414a34-ea2e-4ddf-b425-d0d4bfcacao0,
GrantScope=s3://DOC-EXAMPLE-BUCKET/prefixB*
)
]
)
```

Eliminazione di una concessione

Puoi eliminare un'istanza di Amazon S3 Access Grants da una nel tuo account. L'eliminazione di una concessione di accesso non può essere annullata. Dopo aver eliminato una concessione di accesso, l'assegnatario non avrà più accesso ai tuoi dati Amazon S3.

Puoi eliminare una concessione di accesso utilizzando la console Amazon S3, AWS Command Line Interface (AWS CLI), l'API REST di Amazon S3 e gli SDK. AWS

Utilizzo della console S3

Per eliminare un'autorizzazione di accesso

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Access Grants.
3. Nella pagina S3 Access Grants, scegli la regione che contiene l'istanza S3 Access Grants con cui vuoi lavorare.
4. Scegli Visualizza dettagli per l'istanza.
5. Nella pagina Dettagli, scegli la scheda Concessioni.
6. Cerca la concessione che intendi eliminare. Quando trovi la concessione, scegli il pulsante di opzione accanto a essa.
7. Scegli Elimina. Viene visualizzata una finestra di dialogo che avverte che questa azione non può essere annullata. Scegli nuovamente Elimina per eliminare la concessione.

Usando il AWS CLI

Per installare AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Per utilizzare il seguente comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

Example – Eliminazione di una concessione di accesso

```
aws s3control delete-access-grant \
--account-id 111122223333 \
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

// No response body
```

Utilizzo di REST API

Per informazioni sul supporto REST API di Amazon S3 per la gestione delle concessioni di accesso, consulta DeleteAccessGrant nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Utilizzo degli AWS SDK

Questa sezione fornisce esempi di come eliminare una concessione di accesso utilizzando gli AWS SDK. Per utilizzare il seguente esempio, sostituisci *`user input placeholders`* con le tue informazioni.

Java

Example – Eliminazione di una concessione di accesso

```
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);
}
```

Risposta:

```
DeleteAccessGrantResponse()
```

Richiedi l'accesso ai dati di Amazon S3 tramite S3 Access Grants

Dopo aver utilizzato Amazon S3 Access Grants per creare una concessione di accesso che consenta ai responsabili AWS Identity and Access Management (IAM), alle identità delle directory aziendali o alle applicazioni autorizzate di accedere ai dati S3, i beneficiari possono richiedere le credenziali per accedere a questi dati.

Quando un'applicazione o Servizio AWS utilizza l'operazione `GetDataAccess` API per chiedere a S3 Access Grants l'accesso ai tuoi dati S3 per conto di un beneficiario, S3 Access Grants verifica innanzitutto che tu abbia concesso a questa identità l'accesso ai dati. Quindi, S3 Access Grants utilizza l'operazione AssumeRoleAPI per ottenere un token di credenziali temporaneo e lo invia al richiedente. Questo token di credenziali temporaneo è un token AWS Security Token Service (AWS STS).

La richiesta `GetDataAccess` deve includere il parametro `target`, che specifica l'ambito dei dati S3 a cui si applicano le credenziali temporanee. Questo ambito `target` può essere lo stesso dell'ambito

della concessione o di un sottoinsieme di tale ambito, ma l'ambito `target` deve rientrare nell'ambito della concessione accordata al richiedente. La richiesta deve inoltre specificare il parametro `permission` per indicare il livello di autorizzazione per le credenziali temporanee, `READ`, `WRITE` o `READWRITE`.

Il richiedente può specificare il livello di privilegio del token temporaneo nella richiesta di credenziali. Utilizzando il parametro `privilege`, il richiedente può ridurre o ingrandire l'ambito di accesso delle credenziali temporanee entro i limiti dell'ambito della concessione. Il valore predefinito del parametro `privilege` è `Default`, il che significa che l'ambito di destinazione della credenziale restituita è l'ambito della concessione originale. L'altro valore possibile per `privilege` è `Minimal`. Se l'ambito `target` viene ridotto rispetto all'ambito della concessione originale, la credenziale temporanea viene ridimensionata per corrispondere all'ambito `target`, purché l'ambito `target` rientri nell'ambito della concessione.

La tabella seguente descrive in dettaglio l'effetto del parametro `privilege` su due concessioni. Una concessione ha l'ambito `S3://`*`example-s3-bucket1`*`/bob/*`, che include l'intero prefisso `bob/` nel bucket *`example-s3-bucket1`*. Una concessione ha l'ambito `S3://`*`example-s3-bucket1`*`/bob/reports/*`, che include l'intero prefisso `bob/reports/` nel bucket *`example-s3-bucket1`*.

Ambito della concessione	Ambito richiesto	Privilegio	Ambito restituito	Effetto
`S3://`*`example-s3-bucket1`*`/bob/*`	*`example-s3-bucket1`*`/bob/*`	`Default`	*`example-s3-bucket1`*`/bob/*`	Il richiedente ha accesso a tutti gli oggetti i cui nomi della chiave iniziano con il prefisso *`bob/`* nel bucket *`example-s3-bucket1`*.
`S3://`*`example-s3-bucket1`*`/bob/*`	*`example-s3-bucket1`*`/bob/`	`Minimal`	*`example-s3-bucket1`*`/bob/`	Senza un carattere jolly * dopo il nome del prefisso `bob/`, il richiedente ha accesso solo all'oggetto denominato `bob/` nel bucket *`example-s3-bucket1`*. Non è

Ambito della concessione	Ambito richiesto	Privilegio	Ambito restituito	Effetto
				comune avere un oggetto del genere. Il richiedente non ha accesso a nessun altro oggetto, compresi quelli con nomi chiave che iniziano con il prefisso bob/.
S3://*example-s3-bucket1* /bob/*	*example-s3-bucket1* /bob/images/*	Minimal	*example-s3-bucket1* /bob/images/*	Il richiedente ha accesso a tutti gli oggetti i cui nomi della chiave iniziano con il prefisso *bob/images/** nel bucket *example-s3-bucket1* .
S3://*example-s3-bucket1* /bob/reports/*	*example-s3-bucket1* /bob/reports/file.txt	Default	*example-s3-bucket1* /bob/reports/*	Il richiedente ha accesso a tutti gli oggetti i cui nomi della chiave iniziano con il prefisso bob/reports nel bucket *example-s3-bucket1* , che è l'ambito della concessione corrispondente.

Ambito della concessione	Ambito richiesto	Privilegio	Ambito restituito	Effetto
S3://*example-s3-bucket1* /bob/reports/*	*example-s3-bucket1* /bob/reports/file.txt	Minimal	*example-s3-bucket1* /bob/reports/file.txt	Il richiedente ha accesso solo all'oggetto con il nome della chiave bob/reports/file.txt nel bucket *example-s3-bucket1* . Il richiedente non ha accesso a nessun altro oggetto.

Il parametro `durationSeconds` imposta la durata della credenziale temporanea, in secondi. Il valore predefinito è `3600` secondi (1 ora), ma il richiedente (l'assegnatario) può specificare un intervallo da `900` secondi (15 minuti) a `43200` secondi (12 ore). Se l'assegnatario richiede un valore superiore a questo valore massimo, la richiesta ha esito negativo.

> **Note**
>
> Nella richiesta di un token temporaneo, se la posizione è un oggetto, imposta il valore del parametro `targetType` nella richiesta a `Object`. Questo parametro è obbligatorio solo se la posizione è un oggetto e il livello di privilegio è `Minimal`. Se la posizione è un bucket o un prefisso, non devi specificare questo parametro.

Per ulteriori informazioni, consulta GetDataAccessAmazon Simple Storage Service API Reference.

Puoi richiedere credenziali temporanee utilizzando AWS Command Line Interface (AWS CLI), l'API REST di Amazon S3 e AWS gli SDK.

Usando il AWS CLI

Per installare AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Per utilizzare il seguente comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

Example Richiesta di credenziali temporanee

Richiesta:

```
aws s3control get-data-access \
--account-id 111122223333 \
--target s3://example-s3-bucket/prefixA* \
--permission READ \
--privilege Default \
--region us-east-2
```

Risposta:

```
{
"Credentials": {
 "AccessKeyId": "Example-key-id",
 "SecretAccessKey": "Example-access-key",
 "SessionToken": "Example-session-token",
"Expiration": "2023-06-14T18:56:45+00:00"},
    "MatchedGrantTarget": "s3://example-s3-bucket/prefixA**"
}
```

Utilizzo di REST API

Per informazioni sul supporto dell'API REST di Amazon S3 per la richiesta di credenziali temporanee da S3 Access Grants, consulta il riferimento all'API di Amazon GetDataAccessSimple Storage Service.

Utilizzo degli SDK AWS

Questa sezione fornisce un esempio di come i beneficiari richiedono credenziali temporanee a S3 Access Grants utilizzando gli SDK. AWS

Java

> Il seguente esempio di codice restituisce le credenziali temporanee utilizzate dall'assegnatario per accedere ai tuoi dati S3. Per utilizzare questo esempio di codice, sostituisci *user input placeholders* con le tue informazioni.

Example Ottenimento di credenziali temporanee

Richiesta:

```
public void getDataAccess() {
GetDataAccessRequest getDataAccessRequest = GetDataAccessRequest.builder()
.accountId("111122223333")
.permission(Permission.READ)
.privilege(Privilege.MINIMAL)
.target("s3://example-s3-bucket/prefixA*")
.build();
GetDataAccessResponse getDataAccessResponse =
 s3Control.getDataAccess(getDataAccessRequest);
LOGGER.info("GetDataAccessResponse: " + getDataAccessResponse);
}
```

Risposta:

```
GetDataAccessResponse(
Credentials=Credentials(
AccessKeyId="Example-access-key-id",
SecretAccessKey="Example-secret-access-key",
SessionToken="Example-session-token",
Expiration=2023-06-07T06:55:24Z
))
```

Accedi ai dati S3 tramite una concessione di accesso

Dopo aver ottenuto le credenziali temporanee tramite la concessione di accesso, un assegnatario può utilizzare tali credenziali per chiamare le operazioni API di Amazon S3 per accedere ai tuoi dati.

I beneficiari possono accedere ai dati S3 utilizzando AWS Command Line Interface (AWS CLI), gli AWS SDK e l'API REST di Amazon S3.

Usando il AWS CLI

Dopo aver ottenuto le credenziali temporanee da S3 Access Grants, l'assegnatario può configurare un profilo con tali credenziali per richiamare i dati.

Per installare AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Per utilizzare i seguenti comandi di esempio, sostituisci *user input placeholders* con le tue informazioni.

Example – Configura un profilo

```
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
```

Per utilizzare il seguente comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

Example – Ottieni i dati S3

Il beneficiario può utilizzare il get-object AWS CLI comando per accedere ai dati. Il beneficiario può anche utilizzare put-objectls, e altri comandi S3. AWS CLI

```
aws s3api get-object \
--bucket example-s3-bucket1 \
--key myprefix \
--region us-east-2 \
--profile access-grants-consumer-access-profile
```

Utilizzo degli SDK AWS

Questa sezione fornisce esempi di come gli assegnatari possono accedere ai dati S3 utilizzando gli AWS SDK.

Java

> Per esempi su come ottenere dati S3 utilizzando credenziali temporanee, consulta come ottenere un oggetto utilizzando gli AWS SDK e gli esempi di codice Amazon S3 per. AWS SDK for Java 2.x

Accesso multi-account S3 Access Grants

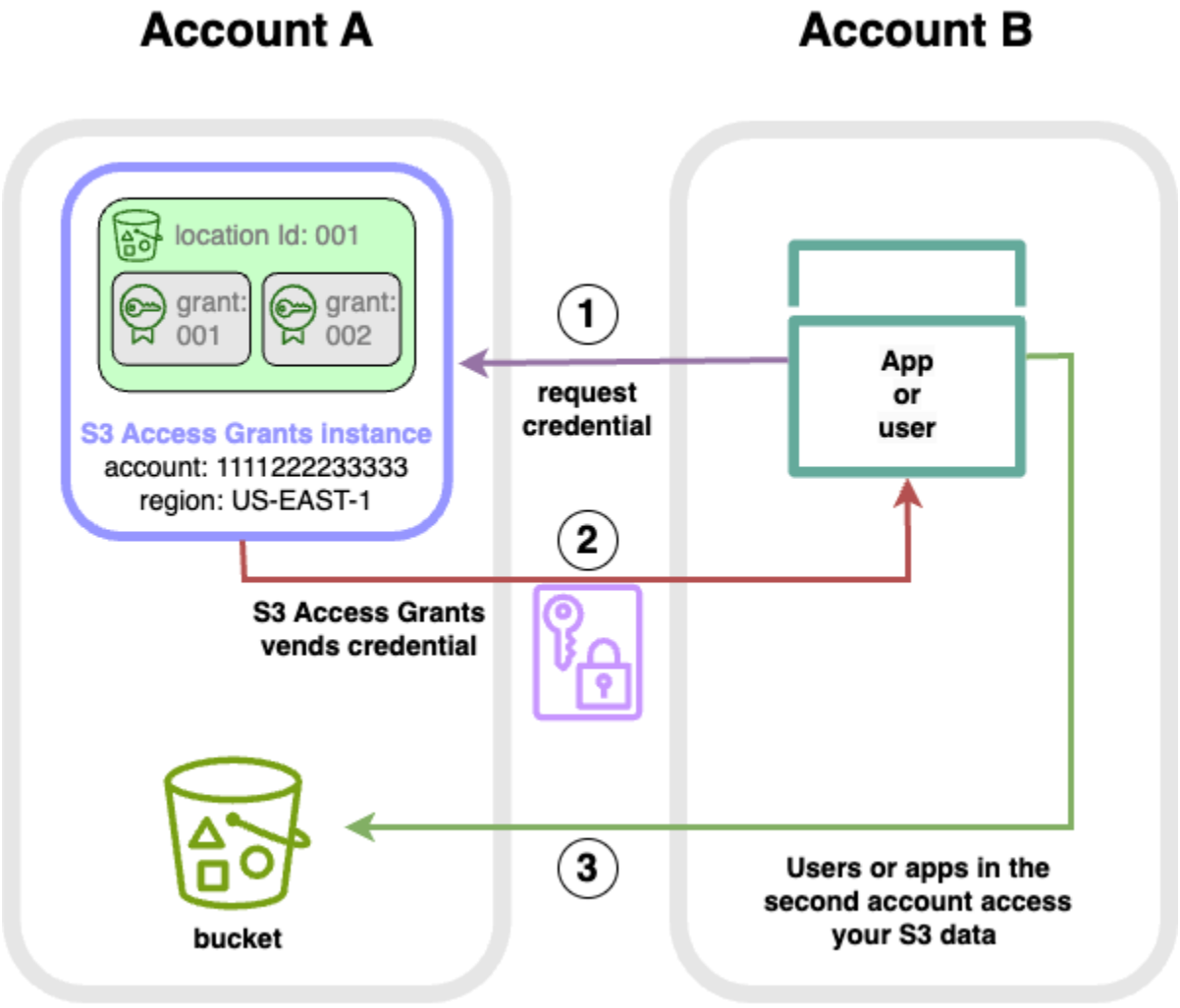
Con S3 Access Grants, puoi concedere l'accesso ai dati di Amazon S3 a quanto segue:

- AWS Identity and Access Management identità (IAM) all'interno del tuo account

- identità IAM in altri account AWS
- Utenti o gruppi di elenchi nella tua istanza AWS IAM Identity Center

Innanzitutto, configura l'accesso tra account per l'altro account. Ciò include la concessione dell'accesso alla tua istanza S3 Access Grants utilizzando una politica delle risorse. Quindi, concedi l'accesso ai tuoi dati S3 (bucket, prefissi o oggetti) utilizzando le concessioni.

Dopo aver configurato l'accesso tra account, l'altro account può richiedere credenziali di accesso temporanee ai dati Amazon S3 da S3 Access Grants. L'immagine seguente mostra il flusso di utenti per l'accesso a S3 tra account diversi tramite S3 Access Grants:



1. Gli utenti o le applicazioni in un secondo account (B) richiedono le credenziali dall'istanza S3 Access Grants nel tuo account (A), dove sono archiviati i dati di Amazon S3. Per ulteriori informazioni, consulta Richiedi l'accesso ai dati di Amazon S3 tramite S3 Access Grants.

2. L'istanza S3 Access Grants nel tuo account (A) restituisce credenziali temporanee se esiste una concessione che consente al secondo account di accedere ai tuoi dati Amazon S3. Per ulteriori informazioni, consulta the section called “Creazione di concessioni”.
3. Gli utenti o le applicazioni del secondo account (B) utilizzano le credenziali fornite da S3 Access Grants per accedere ai dati S3 nel tuo account (A).

La configurazione di S3 Access concede l'accesso a più account

Per concedere l'accesso a S3 su più account tramite S3 Access Grants, segui questi passaggi:

- Passaggio 1: configura un'istanza di S3 Access Grants nel tuo account, ad esempio l'ID `111122223333` dell'account, in cui sono archiviati i dati S3.
- Passaggio 2: configura la politica delle risorse per l'istanza S3 Access Grants nel tuo account `111122223333` per consentire l'accesso al secondo account, ad esempio l'ID dell'account. `444455556666`
- Passaggio 3: configura le autorizzazioni IAM per l'IAM Principal nel secondo account `444455556666` per richiedere le credenziali dall'istanza S3 Access Grants del tuo account. `111122223333`
- Passaggio 4: crea una concessione nel tuo account `111122223333` che consenta all'IAM Principal del secondo account di `444455556666` accedere ad alcuni dati S3 del tuo account. `111122223333`

Passaggio 1: configura un'istanza S3 Access Grants nel tuo account

Innanzitutto, devi avere un'istanza S3 Access Grants nel tuo account `111122223333` per gestire l'accesso ai tuoi dati Amazon S3. Devi creare un'istanza S3 Access Grants in ciascuna delle Regione AWS quali sono archiviati i dati S3 che desideri condividere. Se condividi dati in più di un'unità Regione AWS, ripeti ciascuno di questi passaggi di configurazione per ciascuno. Regione AWS Se hai già un'istanza S3 Access Grants nel luogo in Regione AWS cui sono archiviati i dati S3, procedi al passaggio successivo. Se non hai configurato un'istanza S3 Access Grants, consulta Creazione di un'istanza S3 Access Grants per completare questo passaggio.

Passaggio 2: configura la politica delle risorse per la tua istanza S3 Access Grants per concedere l'accesso su più account

Dopo aver creato un'istanza S3 Access Grants nel tuo account `111122223333` per l'accesso tra account, configura la politica basata sulle risorse per l'istanza S3 Access Grants nel tuo account per

concedere l'accesso a più account. `111122223333` L'istanza S3 Access Grants supporta da sola le policy basate sulle risorse. Con la politica corretta basata sulle risorse, puoi concedere l'accesso alla tua istanza S3 Access Grants a utenti AWS Identity and Access Management (IAM) o ruoli di altri utenti. Account AWS L'accesso tra account diversi concede solo queste autorizzazioni (azioni):

- `s3:GetAccessGrantsInstanceForPrefix`— l'utente, il ruolo o l'app possono recuperare l'istanza S3 Access Grants che contiene un particolare prefisso.
- `s3:ListAccessGrants`
- `s3:ListAccessLocations`
- `s3:GetDataAccess`— l'utente, il ruolo o l'app possono richiedere credenziali temporanee in base all'accesso che ti è stato concesso tramite S3 Access Grants. Usa queste credenziali per accedere ai dati S3 a cui ti è stato concesso l'accesso.

Puoi scegliere quali di queste autorizzazioni includere nella policy della risorsa. Questa politica delle risorse sull'istanza S3 Access Grants è una normale politica basata sulle risorse e supporta tutto ciò che supporta il linguaggio di policy IAM. Nella stessa policy, puoi concedere l'accesso a identità IAM specifiche nel tuo account`111122223333`, ad esempio, utilizzando la `aws:PrincipalArn` condizione, ma non devi farlo con S3 Access Grants. Invece, all'interno dell'istanza S3 Access Grants, puoi creare sovvenzioni per identità IAM individuali dal tuo account e per l'altro account. Gestendo ogni concessione di accesso tramite S3 Access Grants, puoi ridimensionare le tue autorizzazioni.

Se utilizzi già AWS Resource Access Manager(AWS RAM), puoi usarlo per condividere `s3:AccessGrants`le tue risorse con altri account o all'interno della tua organizzazione. Per ulteriori informazioni, consulta Lavorare con AWS risorse condivise. Se non la utilizzi AWS RAM, puoi anche aggiungere la politica delle risorse utilizzando le operazioni dell'API S3 Access Grants o il AWS Command Line Interface ()AWS CLI.

Utilizzo della console S3

Ti consigliamo di utilizzare la console AWS Resource Access Manager (AWS RAM) per condividere `s3:AccessGrants` le tue risorse con altri account o all'interno della tua organizzazione. Per condividere S3 Access Grants su più account, procedi come segue:

Per configurare la politica delle risorse dell'istanza S3 Access Grants:

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

2. Seleziona il Regione AWS dal Regione AWS selettore.
3. Dal riquadro di navigazione a sinistra, seleziona Access Grants.
4. Nella pagina dell'istanza di Access Grants, nella sezione Istanza in questo account, seleziona Condividi istanza. Questo ti reindirizzerà alla AWS RAM console.
5. Seleziona Crea condivisione di risorse.
6. Segui i AWS RAM passaggi per creare la condivisione di risorse. Per ulteriori informazioni, vedere Creazione di una condivisione di risorse in AWS RAM.

Utilizzando il AWS CLI

Per installare AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

È possibile aggiungere la politica delle risorse utilizzando il comando `put-access-grants-instance-resource-policy` CLI.

Se desideri concedere l'accesso multiaccount all'istanza di S3 Access Grants presente nel tuo account `111122223333` al secondo account`444455556666`, la politica delle risorse per l'istanza S3 Access Grants del tuo account `111122223333` dovrebbe `444455556666` autorizzare il secondo account a eseguire le seguenti azioni:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Nella politica delle risorse dell'istanza S3 Access Grants, specifica l'ARN della tua istanza S3 Access Grants come e il secondo account `Resource` come. `444455556666` `Principal` Per utilizzare l'esempio seguente, sostituisci i *`segnaposto di input dell'utente`* con le tue informazioni.

```
{
"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"
} ]
}
```

Per aggiungere o aggiornare la politica delle risorse dell'istanza S3 Access Grants, usa il comando seguente. Quando usi il comando di esempio seguente, sostituiscilo *user input placeholders* con le tue informazioni.

Example Aggiungi o aggiorna la politica delle risorse dell'istanza 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 Ottieni una policy sulla risorsa S3 Access Grants

Puoi anche utilizzare la CLI per ottenere o eliminare una politica delle risorse per un'istanza S3 Access Grants.

Per ottenere una politica delle risorse di S3 Access Grants, usa il seguente comando di esempio. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
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 Elimina una policy della risorsa S3 Access Grants

Per eliminare una politica delle risorse di S3 Access Grants, usa il seguente comando di esempio. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
aws s3control delete-access-grants-instance-resource-policy \
--account-id 111122223333 \
--region us-east-2

// No response body
```

Utilizzo di REST API

Puoi aggiungere la politica delle risorse utilizzando l'PutAccessGrantsInstanceResourcePolicy API.

Se desideri concedere l'accesso su più account all'istanza di S3 Access Grants presente nel tuo account 111122223333 al secondo account444455556666, la politica delle risorse per l'istanza S3 Access Grants del tuo account 111122223333 dovrebbe 444455556666 autorizzare il secondo account a eseguire le seguenti azioni:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Nella politica delle risorse dell'istanza S3 Access Grants, specifica l'ARN della tua istanza S3 Access Grants come e il secondo account `Resource` come. `444455556666` `Principal` Per utilizzare l'esempio seguente, sostituisci i *`segnaposto di input dell'utente`* con le tue informazioni.

```
{
"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"
} ]
}
```

È quindi possibile utilizzare l'PutAccessGrantsInstanceResourcePolicy API per configurare la politica.

Per informazioni sul supporto dell'API REST per aggiornare, ottenere o eliminare una politica delle risorse per un'istanza S3 Access Grants, consulta le seguenti sezioni nel riferimento all'API di Amazon Simple Storage Service:

- PutAccessGrantsInstanceResourcePolicy
- GetAccessGrantsInstanceResourcePolicy
- DeleteAccessGrantsInstanceResourcePolicy

Utilizzo degli SDK AWS

Questa sezione fornisce esempi AWS SDK su come configurare la politica delle risorse di S3 Access Grants per concedere a un secondo AWS account l'accesso ad alcuni dei tuoi dati S3.

Java

Aggiungi, aggiorna, ottieni o elimina una policy della risorsa per gestire l'accesso multi-account alla tua istanza S3 Access Grants.

Example Aggiungi o aggiorna una policy sulle risorse dell'istanza S3 Access Grants

Se desideri concedere l'accesso multiaccount all'istanza di S3 Access Grants presente nel tuo account `111122223333` al secondo account`444455556666`, la politica delle risorse per l'istanza S3 Access Grants del tuo account `111122223333` dovrebbe autorizzare il secondo account `444455556666` a eseguire le seguenti azioni:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Nella politica delle risorse dell'istanza S3 Access Grants, specifica l'ARN della tua istanza S3 Access Grants come e il secondo account `Resource` come. `444455556666` `Principal` Per utilizzare l'esempio seguente, sostituisci i *`segnaposto di input dell'utente`* con le tue informazioni.

```
{
"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"
} ]
}
```

Per aggiungere o aggiornare una politica delle risorse dell'istanza S3 Access Grants, usa il seguente esempio di codice:

```
public void putAccessGrantsInstanceResourcePolicy() {
 PutAccessGrantsInstanceResourcePolicyRequest putRequest =
 PutAccessGrantsInstanceResourcePolicyRequest.builder()
 .accountId(111122223333)
 .policy(RESOURCE_POLICY)
 .build();
 PutAccessGrantsInstanceResourcePolicyResponse putResponse =
 s3Control.putAccessGrantsInstanceResourcePolicy(putRequest);
 LOGGER.info("PutAccessGrantsInstanceResourcePolicyResponse: " + putResponse);
 }
```

Risposta:

```
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 Ottieni una policy sulla risorsa S3 Access Grants

Per ottenere una politica delle risorse di S3 Access Grants, usa il seguente esempio di codice. Per utilizzare il seguente comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
public void getAccessGrantsInstanceResourcePolicy() {
 GetAccessGrantsInstanceResourcePolicyRequest getRequest =
 GetAccessGrantsInstanceResourcePolicyRequest.builder()
 .accountId(111122223333)
 .build();
 GetAccessGrantsInstanceResourcePolicyResponse getResponse =
 s3Control.getAccessGrantsInstanceResourcePolicy(getRequest);
 LOGGER.info("GetAccessGrantsInstanceResourcePolicyResponse: " + getResponse);
 }
```

Risposta:

```
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 Elimina una policy della risorsa S3 Access Grants

Per eliminare una policy relativa alle risorse di S3 Access Grants, usa il seguente esempio di codice. Per utilizzare il seguente comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
public void deleteAccessGrantsInstanceResourcePolicy() {
 DeleteAccessGrantsInstanceResourcePolicyRequest deleteRequest =
 DeleteAccessGrantsInstanceResourcePolicyRequest.builder()
 .accountId(111122223333)
 .build();
 DeleteAccessGrantsInstanceResourcePolicyResponse deleteResponse =
 s3Control.putAccessGrantsInstanceResourcePolicy(deleteRequest);
 LOGGER.info("DeleteAccessGrantsInstanceResourcePolicyResponse: " + deleteResponse);
 }
```

Risposta:

```
DeleteAccessGrantsInstanceResourcePolicyResponse()
```

Passaggio 3: concedi alle identità IAM in un secondo account l'autorizzazione a chiamare l'istanza S3 Access Grants nel tuo account

Dopo che il proprietario dei dati di Amazon S3 ha configurato la policy cross-account per l'istanza S3 Access Grants nell'account111122223333, il proprietario del secondo account 444455556666 deve creare una policy basata sull'identità per i suoi utenti o ruoli IAM e il proprietario deve concedere loro l'accesso all'istanza S3 Access Grants. Nella politica basata sull'identità, includi una o più delle seguenti azioni, a seconda di ciò che è concesso nella politica delle risorse dell'istanza S3 Access Grants e delle autorizzazioni che desideri concedere:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Seguendo il modello di accesso AWS tra account, gli utenti o i ruoli IAM nel secondo account 444455556666 devono disporre esplicitamente di una o più di queste autorizzazioni. Ad esempio, concedi l'`s3:GetDataAccess`autorizzazione in modo che l'utente o il ruolo IAM possa chiamare l'istanza S3 Access Grants nell'account per richiedere le credenziali. 111122223333

Per utilizzare questo comando di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
  "Effect": "Allow",
  "Action": [
   "s3:GetDataAccess",
  ],
   "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
  }
 ]
```

```
}
```

Per informazioni sulla modifica delle policy basate sull'identità IAM, consulta Modifica delle policy IAM nella guida.AWS Identity and Access Management

Passaggio 4: crea una concessione nell'istanza S3 Access Grants del tuo account che consenta all'identità IAM del secondo account di accedere ad alcuni dei tuoi dati S3

Per la fase finale di configurazione, puoi creare una concessione nell'istanza S3 Access Grants del tuo account 111122223333 che dia accesso all'identità IAM nel secondo account 444455556666 ad alcuni dati S3 del tuo account. Puoi farlo utilizzando la console Amazon S3, la CLI, l'API e gli SDK. Per ulteriori informazioni, consulta Creazione di concessioni.

Nella concessione, specifica l' AWS ARN dell'identità IAM del secondo account e specifica a quale posizione nei dati S3 (un bucket, un prefisso o un oggetto) a cui concedi l'accesso. Questa posizione deve essere già registrata con l'istanza S3 Access Grants. Per ulteriori informazioni, consulta Registrazione di una posizione. Facoltativamente, puoi specificare un sottoprefisso. Ad esempio, se la posizione a cui concedi l'accesso è un bucket e desideri limitare ulteriormente l'accesso a un oggetto specifico in quel bucket, passa il nome della chiave dell'oggetto nel campo. `S3SubPrefix` Oppure, se desideri limitare l'accesso agli oggetti nel bucket con nomi di chiave che iniziano con un prefisso specifico, ad esempio, then pass. `2024-03-research-results/` `S3SubPrefix=2024-03-research-results/`

Di seguito è riportato un esempio di comando CLI per creare una concessione di accesso per un'identità nel secondo account. Per ulteriori informazioni, consulta Creazione di concessioni. Per utilizzare questo comando di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

```
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
```

Dopo aver configurato l'accesso tra più account, l'utente o il ruolo nel secondo account può eseguire le seguenti operazioni:

- Chiamate `ListAccessGrantsInstances` per elencare le istanze di S3 Access Grants condivise con esso. AWS RAM Per ulteriori informazioni, consulta Visualizza i dettagli di un'istanza S3 Access Grants.
- Richiede credenziali temporanee da S3 Access Grants. Per ulteriori informazioni su come effettuare queste richieste, consulta. Richiedi l'accesso ai dati di Amazon S3 tramite S3 Access Grants

Utilizzo dei AWS tag con S3 Access Grants

I tag in Amazon S3 Access Grants hanno caratteristiche simili ai tag degli oggetti in Amazon S3. Ogni tag è una coppia chiave-valore. Le risorse in S3 Access Grants alle quali puoi aggiungere tag sono istanze, posizioni e concessioni di S3 Access Grants.

> **Note**
>
> L'assegnazione di tag in S3 Access Grants utilizza operazioni API diverse rispetto all'assegnazione di tag agli oggetti. S3 Access Grants utilizza le operazioni API TagResource, UntagResource e ListTagsForResource in cui una risorsa può essere una posizione registrata, una concessione di accesso o un'istanza S3 Access Grants.

Analogamente ai tag degli oggetti, si applicano le seguenti limitazioni:

- Puoi aggiungere tag alle nuove risorse S3 Access Grants al momento della loro creazione oppure puoi aggiungere tag alle risorse esistenti.
- Puoi associare fino a un massimo di 10 tag a ciascuna risorsa. Se alla stessa risorsa sono associati più tag, questi devono avere chiavi di tag univoche.
- Una chiave di tag può essere composta da un massimo di 128 caratteri Unicode e i valori di tag possono essere composti da un massimo di 256 caratteri Unicode. I tag sono rappresentati internamente in UTF-16. In UTF-16, i caratteri utilizzano 1 o 2 posizioni carattere.
- Per chiavi e valori viene fatta distinzione tra maiuscole e minuscole.

Per ulteriori informazioni sulle restrizioni sui tag, consulta Restrizioni sui tag definiti dall'utente nella Guida per l'utente di AWS Billing .

Puoi taggare le risorse in S3 Access Grants utilizzando AWS Command Line Interface (AWS CLI), l'API REST di Amazon S3 o gli SDK. AWS

Usando il AWS CLI

Per installare AWS CLI, vedere Installazione di AWS CLI nella Guida per l'AWS Command Line Interface utente.

Puoi aggiungere tag a una risorsa S3 Access Grants quando la crei o dopo averla creata. Di seguito sono riportati esempi che mostrano come aggiungere tag a un'istanza S3 Access Grants o rimuoverli da essa. Puoi eseguire operazioni simili per le posizioni registrate e le concessioni di accesso.

Per utilizzare i seguenti comandi di esempio, sostituisci *user input placeholders* con le tue informazioni.

Example – Crea un'istanza S3 Access Grants con tag

```
aws s3control create-access-grants-instance \
 --account-id 111122223333 \
 --profile access-grants-profile \
 --region us-east-2 \
 --tags Key=tagKey1,Value=tagValue1
```

Risposta:

```
 {
    "CreatedAt": "2023-10-25T01:09:46.719000+00:00",
    "AccessGrantsInstanceId": "default",
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default"
}
```

Example – Aggiungi un tag a un'istanza S3 Access Grants già creata

```
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 – Elenca i tag per l'istanza 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
```

Risposta:

```
{
    "Tags": [
        {
            "Key": "tagKey1",
            "Value": "tagValue1"
        },
        {
            "Key": "tagKey2",
            "Value": "tagValue2"
        }
    ]
}
```

Example – Rimuovi tag dall'istanza 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"
```

Utilizzo di REST API

Puoi utilizzare l'API Amazon S3 per aggiungere o rimuovere i tag oppure per elencare i tag per una posizione registrata, una concessione di accesso o un'istanza S3 Access Grants. Per informazioni sul supporto REST API di Amazon S3 per la gestione dei tag S3 Access Grants, consulta le sezioni seguenti nella Documentazione di riferimento delle API di Amazon Simple Storage Service:

- TagResource
- UntagResource
- ListTagsForResource

Limitazioni di S3 Access Grants

S3 Access Grants presenta le seguenti limitazioni:

> **Note**
>
> Se il tuo caso d'uso supera queste limitazioni, contatta l' AWS assistenza per richiedere limiti più elevati.

Istanza S3 Access Grants

Puoi creare 1 istanza S3 Access Grants per account. Regione AWS Consulta Creazione di un'istanza S3 Access Grants.

Posizione di S3 Access Grants

Puoi registrare 1.000 posizioni S3 Access Grants per istanza S3 Access Grants. Consulta Registrazione di una posizione S3 Access Grants.

Grant

Puoi creare solo 100.000 concessioni per istanza S3 Access Grants. Consulta Creazione di una concessione.

Integrazioni con S3 Access Grants

S3 Access Grants può essere utilizzato con i seguenti servizi e funzionalità. AWS Questa pagina verrà aggiornata non appena saranno disponibili nuove integrazioni.

AWS IAM Identity Center

Propagazione delle identità attendibili tra le applicazioni

Amazon EMR

Avvio di un cluster Amazon EMR con S3 Access Grants

Amazon EMR su EKS

Avvio di un cluster Amazon EMR su EKS con S3 Access Grants

Applicazione Amazon EMR serverless

Avvio di un'applicazione Amazon EMR serverless con S3 Access Grants

Amazon Athena

- Utilizzo dei gruppi di lavoro Athena abilitati per il Centro identità IAM

Gestione degli accessi con le ACL

Le liste di controllo degli accessi (ACL) sono una delle opzioni basate sulle risorse che puoi utilizzare per gestire l'accesso ai tuoi bucket e oggetti. È possibile utilizzare gli ACL per concedere autorizzazioni di lettura/scrittura di base ad altri. Account AWS Esistono dei limiti alla gestione delle autorizzazioni tramite le ACL.

Ad esempio, puoi concedere autorizzazioni solo ad altri Account AWS; non puoi concedere autorizzazioni agli utenti del tuo account. Non si possono concedere autorizzazioni condizionali e non è possibile rifiutare le autorizzazioni in modo esplicito. Le ACL sono idonee per scenari specifici. Ad esempio, se il proprietario di un bucket consente Account AWS ad altri di caricare oggetti, le autorizzazioni relative a tali oggetti possono essere gestite utilizzando l'ACL dell'oggetto solo dal proprietario dell' Account AWS oggetto.

S3 Proprietà dell'oggetto è un'impostazione a livello di bucket Amazon S3 che è possibile utilizzare per controllare la proprietà degli oggetti caricati nel bucket e per disabilitare o abilitare le liste di controllo degli accessi (ACL). Per impostazione predefinita, Proprietà dell'oggetto è impostata su Proprietario del bucket applicato e tutte le liste di controllo degli accessi (ACL) sono disabilitate. Quando le ACL sono disabilitate, il proprietario del bucket dispone di tutti gli oggetti nel bucket e gestisce l'accesso ad essi in maniera esclusiva utilizzando policy di gestione dell'accesso.

La maggior parte degli attuali casi d'uso in Amazon S3 non richiede più l'uso delle ACL. È consigliabile mantenere le ACL disabilitate, tranne nelle circostanze in cui è necessario controllare individualmente l'accesso per ciascun oggetto. Con le ACL disabilitate, puoi utilizzare le policy per controllare l'accesso a tutti gli oggetti nel bucket, a prescindere da chi ha caricato gli oggetti nel bucket. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

> **Important**
>
> Se il bucket utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto S3, è necessario utilizzare le policy per concedere l'accesso al bucket e agli oggetti in esso contenuti. Con l'impostazione Proprietario del bucket applicato abilitata, le richieste per impostare liste di controllo degli accessi (ACL) e aggiornare le ACL non vanno a buon fine e

restituiscono il codice di errore `AccessControlListNotSupported`. Le richieste di lettura delle ACL sono ancora supportate.

Per ulteriori informazioni sugli ACL, consulta i seguenti argomenti:

Argomenti

- Panoramica delle liste di controllo accessi (ACL)
- Configurazione delle ACL
- Esempi di policy per gli ACL

Panoramica delle liste di controllo accessi (ACL)

Le liste di controllo accessi (ACL) di Amazon S3 permettono di gestire l'accesso ai bucket e agli oggetti. A ogni bucket e oggetto è associata una ACL come sottorisorsa. Definisce a quali Account AWS gruppi è concesso l'accesso e il tipo di accesso. Quando viene ricevuta una richiesta relativa a una risorsa, Amazon S3 controlla la lista ACL corrispondente per verificare che il richiedente disponga delle autorizzazioni di accesso necessarie.

S3 Proprietà dell'oggetto è un'impostazione a livello di bucket Amazon S3 che è possibile utilizzare per controllare la proprietà degli oggetti caricati nel bucket e per disabilitare o abilitare le liste di controllo degli accessi (ACL). Per impostazione predefinita, Proprietà dell'oggetto è impostata su Proprietario del bucket applicato e tutte le liste di controllo degli accessi (ACL) sono disabilitate. Quando le ACL sono disabilitate, il proprietario del bucket dispone di tutti gli oggetti nel bucket e gestisce l'accesso ad essi in maniera esclusiva utilizzando policy di gestione dell'accesso.

La maggior parte degli attuali casi d'uso in Amazon S3 non richiede più l'uso delle ACL. È consigliabile mantenere le ACL disabilitate, tranne nelle circostanze in cui è necessario controllare individualmente l'accesso per ciascun oggetto. Con le ACL disabilitate, puoi utilizzare le policy per controllare l'accesso a tutti gli oggetti nel bucket, a prescindere da chi ha caricato gli oggetti nel bucket. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

> **Important**
>
> Se il bucket utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto S3, è necessario utilizzare le policy per concedere l'accesso al bucket e agli oggetti in esso

contenuti. Con l'impostazione Proprietario del bucket applicato abilitata, le richieste per impostare liste di controllo degli accessi (ACL) e aggiornare le ACL non vanno a buon fine e restituiscono il codice di errore `AccessControlListNotSupported`. Le richieste di lettura delle ACL sono ancora supportate.

Quando crei un bucket o un oggetto, Amazon S3 crea una lista ACL predefinita che concede al proprietario della risorsa il controllo completo su di essa. Questa situazione è illustrata nella seguente ACL del bucket di esempio (l'oggetto ACL predefinito ha la medesima struttura):

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>*** Owner-Canonical-User-ID ***</ID>
    <DisplayName>owner-display-name</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:type="Canonical User">
        <ID>*** Owner-Canonical-User-ID ***</ID>
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

L'ACL di esempio include un elemento `Owner` che identifica il proprietario tramite l'ID utente canonico dell' Account AWS. Per istruzioni sulla ricerca dell'ID utente canonico, consulta Trovare un ID utente Account AWS canonico. L'`Grant`elemento identifica il beneficiario (un gruppo Account AWS o un gruppo predefinito) e l'autorizzazione concessa. Questa ACL predefinita possiede un elemento `Grant` per il proprietario. Per concedere le autorizzazioni aggiungere elementi `Grant`; ognuno di questi elementi identifica l'assegnatario e l'autorizzazione.

> **Note**
>
> Un ACL può avere fino a 100 di questi elementi.

Argomenti

Che cosa si intende per assegnatario?

Un beneficiario può essere uno Account AWS o uno dei gruppi Amazon S3 predefiniti. Concedi l'autorizzazione a Account AWS utilizzare l'indirizzo e-mail o l'ID utente canonico. Se tuttavia specifichi un indirizzo e-mail nella richiesta di concessione, Amazon S3 recupera l'ID utente canonico di tale account e lo aggiunge alla lista ACL. Gli ACL risultanti contengono sempre l'ID utente canonico di Account AWS, non l'indirizzo e-mail di. Account AWS

Quando si concedono i diritti di accesso, si specifica ogni assegnatario come coppia *`type`*`="`*`value`*`"` in cui *`type`* è uno dei seguenti:

- `id`— Se il valore specificato è l'ID utente canonico di un Account AWS
- `uri`: se si concedono autorizzazioni a un gruppo predefinito
- `emailAddress`: se il valore specificato è l'indirizzo e-mail di un Account AWS

> **Important**
>
> L'utilizzo di indirizzi e-mail per specificare un assegnatario è supportato soltanto nelle seguenti Regioni AWS :
>
> - Stati Uniti orientali (Virginia settentrionale)
> - Stati Uniti occidentali (California settentrionale)
> - Stati Uniti occidentali (Oregon)
> - Asia Pacifico (Singapore)
> - Asia Pacifico (Sydney)
> - Asia Pacifico (Tokyo)
> - Europa (Irlanda)

- Sud America (San Paolo)

Per un elenco di tutti gli endpoint e le regioni Amazon S3 supportati, consultare Regioni ed endpoint in Riferimenti generali di Amazon Web Services.

Example Esempio: indirizzo e-mail

Ad esempio, l'`x-amz-grant-read`intestazione seguente concede agli indirizzi e-mail Account AWS identificati dagli indirizzi e-mail l'autorizzazione a leggere i dati degli oggetti e i relativi metadati:

```
x-amz-grant-read: emailAddress="xyz@example.com", emailAddress="abc@example.com"
```

Warning

Quando concedi ad altri Account AWS l'accesso alle tue risorse, tieni presente che Account AWS possono delegare le proprie autorizzazioni agli utenti tramite i propri account. Questa operazione è nota con il nome di accesso multiaccount. Per informazioni sull'utilizzo dell'accesso multiaccount, consulta Creazione di un ruolo per delegare le autorizzazioni a un utente IAM nella Guida per l'utente di IAM.

Trovare un ID utente Account AWS canonico

L'ID utente canonico è associato al tuo Account AWS. Questo ID è una stringa di caratteri lunga, ad esempio:

`79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be`

Per informazioni su come trovare l'ID utente canonico per il tuo account, consulta Trova l'ID utente canonico per il tuo Account AWS nella Guida di riferimento per la gestione dell'account.AWS

Puoi anche cercare l'ID utente canonico di un utente Account AWS leggendo l'ACL di un bucket o di un oggetto a cui dispone delle autorizzazioni di accesso. Account AWS Quando a un individuo Account AWS vengono concesse le autorizzazioni tramite una richiesta di concessione, viene aggiunta una voce di autorizzazione all'ACL con l'ID utente canonico dell'account.

> **Note**
>
> Se rendi pubblico il bucket (non consigliato) qualsiasi utente non autenticato può caricare oggetti nel bucket. Questi utenti anonimi non dispongono di un Account AWS. Quando un utente anonimo carica un oggetto nel tuo bucket Amazon S3 aggiunge un ID utente canonico speciale (`65a011a29cdf8ec533ec3d1ccaae921c`) in quanto proprietario dell'oggetto nell'ACL. Per ulteriori informazioni, consulta Proprietà di bucket e oggetti di Amazon S3.

Gruppi predefiniti di Amazon S3

Amazon S3 include un set di gruppi predefiniti. Quando si concede l'accesso a un gruppo a livello di account, viene specificato uno degli URI Amazon S3 anziché un ID utente canonico. Amazon S3 fornisce i seguenti gruppi predefiniti:

- Gruppo Authenticated Users – Rappresentato da `http://acs.amazonaws.com/groups/global/AuthenticatedUsers`.

  Questo gruppo rappresenta tutto. Account AWSL'autorizzazione di accesso a questo gruppo consente Account AWS a chiunque di accedere alla risorsa. Tuttavia, tutte le richieste devono essere firmate (autenticate).

  > **Warning**
  >
  > Quando concedi l'accesso al gruppo Authenticated Users, qualsiasi utente AWS autenticato al mondo può accedere alla tua risorsa.

- Gruppo All Users – Rappresentato da `http://acs.amazonaws.com/groups/global/AllUsers`.

  L'autorizzazione di accesso per questo gruppo consente a qualsiasi persona al mondo di accedere alla risorsa. Le richieste possono essere firmate (autenticate) o non firmate (anonime). Le richieste non firmate mancano dell'intestazione di autenticazione.

  > **Warning**
  >
  > È vivamente consigliato non concedere mai al gruppo All Users autorizzazioni `WRITE`, `WRITE_ACP` o `FULL_CONTROL`. Ad esempio, mentre le autorizzazioni `WRITE` non consentono ai non proprietari di sovrascrivere o eliminare oggetti esistenti, le autorizzazioni

`WRITE` consentono comunque a chiunque di memorizzare oggetti nel bucket, per i quali vengono fatturati. Per ulteriori dettagli su queste autorizzazioni, consulta la sezione Quali autorizzazioni è possibile concedere?.

- Gruppo Log Delivery – Rappresentato da `http://acs.amazonaws.com/groups/s3/LogDelivery`.

  L'autorizzazione `WRITE` per un bucket consente a questo gruppo di scriver log di credenziali d'accesso al server (consulta Registrazione delle richieste con registrazione dell'accesso al server) per il bucket.

> **Note**
>
> Quando si utilizzano gli ACL, un beneficiario può essere uno Account AWS o uno dei gruppi Amazon S3 predefiniti. Tuttavia, l'assegnatario non può essere un utente IAM. Per ulteriori informazioni sugli utenti AWS e sulle autorizzazioni in IAM, consulta Utilizzo di AWS Identity and Access Management.

Quali autorizzazioni è possibile concedere?

La seguente tabella elenca il set di autorizzazioni che Amazon S3 supporta in una lista ACL. L'insieme di autorizzazioni ACL è lo stesso per le ACL degli oggetti e dei bucket. Tuttavia, a seconda del contesto (bucket ACL o oggetto ACL), queste autorizzazioni si riferiscono a specifiche operazioni sui bucket o sugli oggetti. La tabella elenca le autorizzazioni e ne descrive il significato nel contesto degli oggetti e dei bucket.

Per ulteriori informazioni sulle autorizzazioni ACL nella console di Amazon S3, consulta Configurazione delle ACL.

Autorizzazioni ACL

Autorizzazione	Concessione a livello di bucket	Concessione a livello di oggetto
`READ`	Consente all'assegnatario di elencare gli oggetti del bucket	Consente all'assegnatario di leggere i dati dell'oggetto e i relativi metadata

Autorizzazione	Concessione a livello di bucket	Concessione a livello di oggetto
`WRITE`	Consente all'assegnatario di creare nuovi oggetti del bucket. Per i proprietari di bucket e oggetti di oggetti esistenti, consente anche di eliminare e sovrascrivere tali oggetti.	Non applicabile.
`READ_ACP`	Consente all'assegnatario di leggere l'ACL del bucket	Consente all'assegnatario di leggere l'ACL dell'oggetto
`WRITE_ACP`	Consente all'assegnatario di scrivere l'ACL del bucket interessato	Consente all'assegnatario di scrivere l'ACL dell'oggetto interessato
`FULL_CONTROL`	Consente al beneficiario le autorizzazioni `READ`, `WRITE`, `READ_ACP` e `WRITE_ACP` sul bucket	Consente al beneficiario le autorizzazioni `READ`, `READ_ACP` e `WRITE_ACP` sull'oggetto

> **Warning**
>
> Prestare attenzione a concedere le autorizzazioni di accesso ai bucket e agli oggetti S3. Ad esempio, la concessione dell'accesso `WRITE` a un bucket consente all'assegnatario di creare oggetti nel bucket. È vivamente consigliato di leggere tutta questa sezione Panoramica delle liste di controllo accessi (ACL) prima di concedere autorizzazioni.

Mappatura delle autorizzazioni ACL e delle autorizzazioni della policy di accesso

Come illustrato nella tabella precedente, un'ACL concede solo un insieme finito di autorizzazioni rispetto al numero di autorizzazioni che possono essere definite in una policy d'accesso predefinita (consulta Azioni politiche per Amazon S3). Ognuna di queste autorizzazioni permette di eseguire una o più operazioni di Amazon S3.

La seguente tabella mostra come ogni autorizzazione ACL è mappata sulle autorizzazioni corrispondenti della policy d'accesso predefinita. Come si può vedere, la policy di accesso predefinita concede un numero maggiore di autorizzazioni rispetto all'ACL. Le ACL possono essere utilizzate principalmente per concedere autorizzazioni di base di lettura/scrittura, analogamente alle

autorizzazioni sul file system. Per ulteriori informazioni su quando utilizzare una lista ACL, consulta Identity and Access Management per Amazon S3.

Per ulteriori informazioni sulle autorizzazioni ACL nella console di Amazon S3, consulta Configurazione delle ACL.

Autorizzazione ACL	Autorizzazioni corrispondenti della policy d'accesso predefinita quando l'autorizzazione ACL viene concessa su un bucket	Autorizzazioni corrispondenti della policy d'accesso predefinita quando l'autorizzazione ACL viene concessa su un oggetto
READ	s3:ListBucket , s3:ListBucketVersions e s3:ListBucketMultipartUploads	s3:GetObject e s3:GetObjectVersion
WRITE	s3:PutObject Il proprietario del bucket può creare, sovrascrivere ed eliminare qualsiasi oggetto nel bucket e il proprietario dell'oggetto ha FULL_CONTROL sull'oggetto. Inoltre, quando l'assegnatario è il proprietario del bucket, la concessione dell'autorizzazione WRITE nell'ACL di un bucket consente l'esecuzione dell'operazione s3:DeleteObjectVersion su qualsiasi versione del bucket.	Non applicabile.
READ_ACP	s3:GetBucketAcl	s3:GetObjectAcl e s3:GetObjectVersionAcl
WRITE_ACP	s3:PutBucketAcl	s3:PutObjectAcl e s3:PutObjectVersionAcl
FULL_CONTROL	Equivale alla concessione delle autorizzazioni ACL READ, WRITE,	Equivale alla concessione delle autorizzazioni ACL READ, READ_ACP

Autorizzazione ACL	Autorizzazioni corrispondenti della policy d'accesso predefinita quando l'autorizzazione ACL viene concessa su un bucket	Autorizzazioni corrispondenti della policy d'accesso predefinita quando l'autorizzazione ACL viene concessa su un oggetto
	`READ_ACP` e `WRITE_ACP` . Di conseguenza, questa autorizzazione ACL è mappata su una combinazione delle autorizzazioni corrispondenti della policy d'accesso predefinita.	e `WRITE_ACP` . Di conseguenza, questa autorizzazione ACL è mappata su una combinazione delle autorizza zioni corrispondenti della policy d'accesso predefinita.

Chiavi di condizione

Quando si concedono autorizzazioni per le policy di accesso, è possibile utilizzare le chiavi di condizione per limitare il valore dell'ACL su un oggetto utilizzando una policy del bucket. Le chiavi di contesto riportate di seguito corrispondono alle ACL. È possibile utilizzare queste chiavi di contesto per richiedere l'utilizzo di un'ACL specifica in una richiesta:

- `s3:x-amz-grant-read` – Richiedere l'accesso in lettura.
- `s3:x-amz-grant-write` – Richiedere l'accesso in scrittura.
- `s3:x-amz-grant-read-acp` – Richiedere l'accesso in lettura alla lista ACL del bucket.
- `s3:x-amz-grant-write-acp` - Richiedere l'accesso in scrittura alla lista ACL del bucket.
- `s3:x-amz-grant-full-control` – Richiedere il controllo completo.
- `s3:x-amz-acl` – Richiedere una lista ACL predefinita.

Per policy di esempio con intestazioni specifiche delle liste ACL, consulta Concessione di s3:PutObject autorizzazione con una condizione che richiede al proprietario del bucket di ottenere il pieno controllo. Per un elenco completo delle chiavi di condizione specifiche di Amazon S3, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.

Valori **`aclRequired`** per le richieste di Amazon S3

Per identificare le richieste Amazon S3 che richiedono le ACL per l'autorizzazione, puoi utilizzare il valore `aclRequired` nei log degli accessi del server Amazon S3 oppure AWS CloudTrail. Il `aclRequired` valore visualizzato nei CloudTrail nostri log di accesso al server di Amazon S3 dipende dalle operazioni richiamate e da determinate informazioni sul richiedente, sul proprietario

dell'oggetto e sul proprietario del bucket. Se non era richiesto alcun ACL o se stai impostando l'ACL predefinito o se le richieste sono consentite dalla tua policy del bucket, la stringa di `aclRequired` valore è "" nei log di accesso al server di - Amazon S3 `bucket-owner-full-control` ed è assente in. CloudTrail

Le tabelle seguenti elencano `aclRequired` i valori previsti nei CloudTrail nostri log di accesso al server Amazon S3 per le varie operazioni API di Amazon S3. Puoi utilizzare queste informazioni per capire quali operazioni di Amazon S3 dipendono dagli ACL per l'autorizzazione. Nelle tabelle seguenti, A, B e C rappresentano i diversi account associati al richiedente, al proprietario dell'oggetto e al proprietario del bucket. Le voci con un asterisco (*) indicano uno degli account A, B o C.

> **Note**
>
> Le operazioni `PutObject` nella tabella seguente, se non diversamente specificato, indicano richieste che non impostano un ACL, a meno che l'ACL non sia un `bucket-owner-full-control` ACL. Un valore nullo per `aclRequired` indica che non `aclRequired` è presente nei log. AWS CloudTrail

`aclRequired`valori per CloudTrail

Nome operazione	Richiedente	Proprietario dell'oggetto.	Proprietario del bucket	La politica dei bucket garantisce l'accesso	**`aclRequired`** value	Motivo
`GetObject`	A	A	A	Sì o No	null	Accesso allo stesso account
	A	B	A	Sì o No	null	È stato imposto l'accesso allo stesso account con il proprietario del bucket

Nome operazione	Richiedente	Proprietario dell'oggetto.	Proprietario del bucket	La politica dei bucket garantisce l'accesso	**`aclRequired`** value	Motivo
	A	A	B	Sì	null	Accesso tra account a causa della politica tra account
	A	A	B	No	Sì	L'accesso tra account si basa su ACL
	A	A	B	Sì	null	Accesso tra account a causa della politica tra account
	A	B	B	No	Sì	L'accesso tra account si basa su ACL
	A	B	C	Sì	null	Accesso tra account a causa della politica tra account

Nome operazione	Richiedente	Proprietario dell'oggetto.	Proprietario del bucket	La politica dei bucket garantisce l'accesso	**`aclRequired`** value	Motivo
	A	B	C	No	Sì	L'accesso tra account si basa su ACL
`PutObject`	A	Non applicabile	A	Sì o No	null	Accesso allo stesso account
	A	Non applicabile	B	Sì	null	Accesso tra account a causa della politica tra account
	A	Non applicabile	B	No	Sì	L'accesso tra account si basa su ACL
`PutObject` con un ACL (ad eccezione di `bucket-owner-full-control`)	*	Non applicabile	*	Sì o No	Sì	Richiesta di autorizzazioni ACL
`ListObjects`	A	Non applicabile	A	Sì o No	null	Accesso allo stesso account

Nome operazione	Richiedente	Proprietario dell'oggetto.	Proprietario del bucket	La politica dei bucket garantisce l'accesso	**`aclRequired`** value	Motivo
	A	Non applicabile	B	Sì	null	Accesso tra account a causa della politica tra account
	A	Non applicabile	B	No	Sì	L'accesso tra account si basa su ACL
`DeleteObject`	A	Non applicabile	A	Sì o No	null	Accesso allo stesso account
	A	Non applicabile	B	Sì	null	Accesso tra account a causa della politica tra account
	A	Non applicabile	B	No	Sì	L'accesso tra account si basa su ACL
`PutObjectAcl`	*	*	*	Sì o No	Sì	Richiesta di autorizzazioni ACL

Nome operazione	Richiedente	Proprietario dell'oggetto.	Proprietario del bucket	La politica dei bucket garantisce l'accesso	**aclRequired** value	Motivo
PutBucketAcl	*	Non applicabile	*	Sì o No	Sì	Richiesta di autorizzazioni ACL

> **Note**
>
> Le operazioni REST.PUT.OBJECT nella tabella seguente, se non diversamente specificato, indicano richieste che non impostano un ACL, a meno che l'ACL non sia un bucket-owner-full-control ACL. Una stringa di valori aclRequired di "-" indica un valore nullo nei log di accesso al server Amazon S3.

Valori **aclRequired** per i log di accesso al server Amazon S3

Nome operazione	Richiedente	Proprietario dell'oggetto.	Proprietario del bucket	La politica dei bucket garantisce l'accesso	**aclRequired** value	Motivo
REST.GET.OBJECT	A	A	A	Sì o No	-	Accesso allo stesso account
	A	B	A	Sì o No	-	È stato imposto l'accesso allo stesso account con il proprietario del bucket

Nome operazione	Richiedente	Proprietario dell'oggetto.	Proprietario del bucket	La politica dei bucket garantisce l'accesso	**`aclRequired`** value	Motivo
	A	A	B	Sì	-	Accesso tra account a causa della politica tra account
	A	A	B	No	Sì	L'accesso tra account si basa su ACL
	A	B	B	Sì	-	Accesso tra account a causa della politica tra account
	A	B	B	No	Sì	L'accesso tra account si basa su ACL
	A	B	C	Sì	-	Accesso tra account a causa della politica tra account

Nome operazione	Richiedente	Proprietario dell'oggetto.	Proprietario del bucket	La politica dei bucket garantisce l'accesso	**`aclRequired`** value	Motivo
	A	B	C	No	Sì	L'accesso tra account si basa su ACL
`REST.PUT.OBJECT`	A	Non applicabile	A	Sì o No	-	Accesso allo stesso account
	A	Non applicabile	B	Sì	-	Accesso tra account a causa della politica tra account
	A	Non applicabile	B	No	Sì	L'accesso tra account si basa su ACL
`REST.PUT.OBJECT` con un ACL (ad eccezione di `bucket-owner-full-control`)	*	Non applicabile	*	Sì o No	Sì	Richiesta di autorizzazioni ACL
`REST.GET.BUCKET`	A	Non applicabile	A	Sì o No	-	Accesso allo stesso account

Nome operazione	Richiedente	Proprietario dell'oggetto.	Proprietario del bucket	La politica dei bucket garantisce l'accesso	**`aclRequired`** value	Motivo
	A	Non applicabile	B	Sì	-	Accesso tra account a causa della politica tra account
	A	Non applicabile	B	No	Sì	L'accesso tra account si basa su ACL
`REST.DELETE.OBJECT`	A	Non applicabile	A	Sì o No	-	Accesso allo stesso account
	A	Non applicabile	B	Sì	-	Accesso tra account a causa della politica tra account
	A	Non applicabile	B	No	Sì	L'accesso tra account si basa su ACL
`REST.PUT.ACL`	*	*	*	Sì o No	Sì	Richiesta di autorizzazioni ACL

ACL di esempio

La seguente ACL di esempio su un bucket identifica il proprietario della risorsa e un insieme di concessioni. Il suo formato è la rappresentazione XML di una lista ACL in REST API di Amazon S3. Il proprietario del bucket ha il `FULL_CONTROL` della risorsa. Inoltre, l'ACL mostra come vengono concesse le autorizzazioni su una risorsa a due Account AWS, identificati da un ID utente canonico, e a due dei gruppi Amazon S3 predefiniti discussi nella sezione precedente.

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 predefinita

Amazon S3 supporta un set di concessioni predefinite, dette liste ACL predefinite. Ogni ACL predefinita ha un insieme predefinito di assegnatari e autorizzazioni. La seguente tabella elenca l'insieme di ACL predefinite e le concessioni predefinite associate.

ACL predefinita	Si applica a	Autorizzazioni aggiunte a un'ACL
`private`	Bucket e oggetto	Il proprietario ottiene il `FULL_CONTROL` . Nessun altro ha diritti di accesso (impostazione predefinita).
`public-read`	Bucket e oggetto	Il proprietario ottiene il `FULL_CONTROL` . Il gruppo `AllUsers` (consulta Che cosa si intende per assegnatario?) ottiene l'accesso `READ`.
`public-read-write`	Bucket e oggetto	Il proprietario ottiene il `FULL_CONTROL` . Il gruppo `AllUsers`ottiene l'accesso `READ` e `WRITE`. In genere la concessione di queste autorizzazioni su un bucket non è consigliata.
`aws-exec-read`	Bucket e oggetto	Il proprietario ottiene il `FULL_CONTROL` . Amazon EC2 ottiene l'accesso di tipo `READ` per la richiesta `GET` con

ACL predefinita	Si applica a	Autorizzazioni aggiunte a un'ACL
		cui ottenere un bundle Amazon Machine Image (AMI) da Amazon S3.
`authenticated-read`	Bucket e oggetto	Il proprietario ottiene il `FULL_CONTROL` . Il gruppo `AuthenticatedUsers` ottiene l'accesso `READ`.
`bucket-owner-read`	Oggetto	Il proprietario dell'oggetto ottiene il `FULL_CONTROL` . Il proprietario del bucket ottiene l'accesso `READ`. Se specifichi questa lista ACL predefinita durante la creazione di un bucket, Amazon S3 la ignora.
`bucket-owner-full-control`	Oggetto	Sia il proprietario dell'oggetto che il proprietario del bucket ottengono il `FULL_CONTROL` dell'oggetto. Se specifichi questa lista ACL predefinita durante la creazione di un bucket, Amazon S3 la ignora.
`log-delivery-write`	Bucket	Il gruppo `LogDelivery` ottiene le autorizzazioni `WRITE` e `READ_ACP` sul bucket. Per ulteriori informazioni sui log, consulta (Registrazione delle richieste con registrazione dell'accesso al server).

> **Note**
>
> È possibile specificare solo una di queste ACL predefinite nella richiesta.

Per specificare un'ACL predefinita nella richiesta si utilizza l'intestazione di richiesta `x-amz-acl`. Quando Amazon S3 riceve una richiesta contenente una lista ACL predefinita, aggiunge le concessioni predefinite alla lista ACL della risorsa.

Configurazione delle ACL

Questa sezione descrive come gestire le autorizzazioni di accesso per bucket e oggetti S3 tramite le liste di controllo accessi (ACL). Puoi aggiungere sovvenzioni all'ACL della tua risorsa utilizzando, (AWS Command Line Interface CLI) AWS Management Console, l'API REST o gli SDK. AWS

Le autorizzazioni per il bucket e gli oggetti sono indipendenti l'una dall'altra. Un oggetto non eredita le autorizzazioni dal bucket a cui appartiene. Se ad esempio si crea un bucket e si concede l'accesso in scrittura a un utente, non sarà possibile accedere agli oggetti di tale utente a meno che questi non conceda esplicitamente l'accesso.

Puoi concedere autorizzazioni ad altri Account AWS utenti o a gruppi predefiniti. L'utente o il gruppo a cui si concedono le autorizzazioni è denominato assegnatario. Per impostazione predefinita, il proprietario, che è colui Account AWS che ha creato il bucket, dispone delle autorizzazioni complete.

Ogni autorizzazione concessa a un utente o a un gruppo aggiunge una voce all'ACL associata al bucket. Nell'ACL sono elencate le assegnazioni, che identificano l'assegnatario e l'autorizzazione concessa.

S3 Proprietà dell'oggetto è un'impostazione a livello di bucket Amazon S3 che è possibile utilizzare per controllare la proprietà degli oggetti caricati nel bucket e per disabilitare o abilitare le liste di controllo degli accessi (ACL). Per impostazione predefinita, Proprietà dell'oggetto è impostata su Proprietario del bucket applicato e tutte le liste di controllo degli accessi (ACL) sono disabilitate. Quando le ACL sono disabilitate, il proprietario del bucket dispone di tutti gli oggetti nel bucket e gestisce l'accesso ad essi in maniera esclusiva utilizzando policy di gestione dell'accesso.

La maggior parte degli attuali casi d'uso in Amazon S3 non richiede più l'uso delle ACL. È consigliabile mantenere le ACL disabilitate, tranne nelle circostanze in cui è necessario controllare individualmente l'accesso per ciascun oggetto. Con le ACL disabilitate, puoi utilizzare le policy per controllare l'accesso a tutti gli oggetti nel bucket, a prescindere da chi ha caricato gli oggetti nel bucket. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

> **Important**
>
> Se il bucket utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto S3, è necessario utilizzare le policy per concedere l'accesso al bucket e agli oggetti in esso contenuti. Con l'impostazione Proprietario del bucket applicato abilitata, le richieste per impostare liste di controllo degli accessi (ACL) e aggiornare le ACL non vanno a buon fine e restituiscono il codice di errore `AccessControlListNotSupported`. Le richieste di lettura delle ACL sono ancora supportate.

> **Warning**
>
> Ti consigliamo vivamente di evitare di concedere l'accesso in scrittura ai gruppi Everyone (accesso pubblico) o Authenticated Users (tutti gli utenti AWS autenticati). Per maggiori informazioni sugli effetti della concessione dell'accesso in scrittura a questi gruppi, consulta Gruppi predefiniti di Amazon S3.

Utilizzo della console S3 per impostare le autorizzazioni ACL per un bucket

La console visualizza le concessioni di accesso combinate per gli assegnatari duplicati. Per visualizzare l'elenco completo degli ACL, utilizza l'API REST AWS CLI o gli SDK di Amazon S3. AWS

Nella tabella seguente vengono illustrate le autorizzazioni ACL che è possibile configurare per i bucket nella console di Amazon S3.

Autorizzazioni ACL della console di Amazon S3 per i bucket

Autorizzazione console	Autorizzazione ACL	Accesso
Oggetti – Elenco	`READ`	Consente all'assegnatario di elencare gli oggetti del bucket.
Oggetti - Scrittura	`WRITE`	Consente all'assegnatario di creare nuovi oggetti del bucket. Per i proprietari di bucket e oggetti di oggetti esistenti, consente anche di eliminare e sovrascrivere tali oggetti.
ACL del bucket – Lettura	`READ_ACP`	Consente all'assegnatario di leggere l'ACL del bucket.
ACL del bucket – Scrittura	`WRITE_ACP`	Consente all'assegnatario di scrivere l'ACL del bucket interessato.
Everyone (Tutti) (accesso pubblico)	`READ`	Concede l'accesso pubblico in lettura per gli oggetti nel bucket. Quando si concede l'accesso all'elenco a Everyone (Tutti) (accesso pubblico), chiunque al mondo può accedere agli oggetti nel bucket.

Autorizzazione console	Autorizzazione ACL	Accesso
: Oggetti - Elenco		
Everyone (Tutti) (accesso pubblico) : ACL del bucket - Lettura	`READ_ACP`	Concede l'accesso pubblico in lettura per l'ACL del bucket. Quando si concede l'accesso in lettura a Everyone (Tutti) (accesso pubblico), chiunque al mondo può accedere all'ACL del bucket.

Per ulteriori informazioni sulle autorizzazioni ACL, consulta Panoramica delle liste di controllo accessi (ACL).

> **Important**
>
> Se il bucket utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto S3, è necessario utilizzare le policy per concedere l'accesso al bucket e agli oggetti in esso contenuti. Con l'impostazione Proprietario del bucket applicato abilitata, le richieste per impostare liste di controllo degli accessi (ACL) e aggiornare le ACL non vanno a buon fine e restituiscono il codice di errore `AccessControlListNotSupported`. Le richieste di lettura delle ACL sono ancora supportate.

Come impostare le autorizzazioni ACL per un bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegliere il nome del bucket per cui impostare le autorizzazioni.
3. Seleziona Autorizzazioni.
4. In Lista di controllo degli accessi (ACL), seleziona Modifica.

   Puoi modificare le seguenti autorizzazioni ACL per il bucket:

Oggetti

- List: consente all'assegnatario di elencare gli oggetti nel bucket.
- Scrittura – Consente all'assegnatario di creare nuovi oggetti nel bucket. Per i proprietari di bucket e oggetti di oggetti esistenti, consente anche di eliminare e sovrascrivere tali oggetti.

  Nella console S3, puoi concedere l'accesso in scrittura solo al gruppo di consegna dei log S3 e al proprietario del bucket (il tuo). Account AWS Ti consigliamo vivamente di non concedere l'accesso in scrittura ad altri utenti. Tuttavia, se devi concedere l'accesso in scrittura, puoi utilizzare gli AWS SDK o l' AWS CLI API REST.

ACL del bucket

- Read: consente all'assegnatario di leggere l'ACL del bucket.
- Write: consente all'assegnatario di scrivere l'ACL del bucket interessato.

5. Per modificare le autorizzazioni del proprietario del bucket, oltre a Bucket owner (il tuo Account AWS), deseleziona o seleziona una delle seguenti autorizzazioni ACL:

   - Oggetti – Elenco o scrittura
   - ACL del bucket – Lettura o scrittura

   Il proprietario si riferisce a Utente root dell'account AWS, non a un utente IAM. AWS Identity and Access Management Per ulteriori informazioni sull'utente root, consulta Utente root dell'account AWS nella Guida per l'utente di IAM.

6. Per concedere o annullare le autorizzazioni per il pubblico generale (tutti su Internet), accanto a Tutti (accesso pubblico), deseleziona o seleziona una delle seguenti autorizzazioni ACL:

   - Oggetti – Elenco
   - ACL del bucket – Lettura

   > **Warning**
   >
   > Prestare attenzione nel concedere l'accesso pubblico al bucket S3 al gruppo Everyone (Tutti). Quando si concede l'accesso a questo gruppo, qualsiasi persona al mondo può

accedere al bucket. Si consiglia di non concedere mai alcun tipo di accesso in scrittura pubblico al bucket S3.

7. Per concedere o annullare le autorizzazioni a chiunque disponga di un gruppo Account AWS, oltre al gruppo Authenticated Users (chiunque disponga di un Account AWS), deseleziona o seleziona una delle seguenti autorizzazioni ACL:

   - Oggetti – Elenco
   - ACL del bucket – Lettura

8. Per concedere o annullare le autorizzazioni per Amazon S3 a scrivere i log di accesso al server nel bucket, in gruppo di recapito log S3 deseleziona o seleziona una delle seguenti autorizzazioni ACL:

   - Oggetti – Elenco o scrittura
   - ACL del bucket – Lettura o scrittura

   Se un bucket è configurato come bucket target per la ricezione dei log di accesso, le autorizzazioni del bucket devono permettere al gruppo Log Delivery (Distribuzione log) l'accesso in scrittura al bucket. Quando si abilita la registrazione degli accessi al server in un bucket, la console di Amazon S3 concede l'accesso in scrittura al gruppo Log Delivery (Distribuzione log) per il bucket di destinazione scelto per la ricezione dei log. Per ulteriori informazioni sulla registrazione degli accessi al server, consulta Abilitazione della registrazione degli accessi al server Amazon S3.

9. Per concedere l'accesso a un altro utente, procedi come segue Account AWS:

   a. Scegli Aggiungi assegnatario.
   b. Nella casella Assegnatario, inserisci l'ID canonico dell'altro Account AWS.
   c. Seleziona una delle seguenti autorizzazioni ACL:

      - Oggetti – Elenco o scrittura
      - ACL del bucket – Lettura o scrittura

> ⚠ Warning
>
> Quando concedi ad altri Account AWS l'accesso alle tue risorse, tieni presente che Account AWS possono delegare le proprie autorizzazioni agli utenti tramite i rispettivi account. Questa operazione è nota con il nome di accesso multiaccount. Per

informazioni sull'utilizzo dell'accesso multiaccount, consulta Creazione di un ruolo per delegare le autorizzazioni a un utente IAM nella Guida per l'utente di IAM.

10. Per rimuovere l'accesso a un altro utente Account AWS, in Accesso per altri Account AWS, scegli Rimuovi.
11. Per salvare le modifiche, scegliere Save changes (Salva modifiche).

Utilizzo della console S3 per impostare le autorizzazioni ACL per un oggetto

La console visualizza le concessioni di accesso combinate per gli assegnatari duplicati. Per visualizzare l'elenco completo degli ACL, utilizza l'API REST AWS CLI o gli SDK di Amazon S3. AWS Nella tabella seguente vengono illustrate le autorizzazioni ACL che è possibile configurare per gli oggetti nella console di Amazon S3.

Autorizzazioni ACL della console di Amazon S3 per gli oggetti

Autorizzazione console	Autorizzazione ACL	Accesso
Oggetto - Lettura	`READ`	Consente all'assegnatario di leggere i dati dell'oggetto e i relativi metadati.
ACL dell'oggetto - Lettura	`READ_ACP`	Consente all'assegnatario di leggere l'ACL dell'oggetto.
ACL dell'oggetto - Scrittura	`WRITE_ACP`	Consente all'assegnatario di scrivere l'ACL dell'oggetto interessato

Per ulteriori informazioni sulle autorizzazioni ACL, consulta Panoramica delle liste di controllo accessi (ACL).

Important

Se il bucket utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto S3, è necessario utilizzare le policy per concedere l'accesso al bucket e agli oggetti in esso contenuti. Con l'impostazione Proprietario del bucket applicato abilitata, le richieste per impostare liste di controllo degli accessi (ACL) e aggiornare le ACL non vanno a buon fine e

restituiscono il codice di errore `AccessControlListNotSupported`. Le richieste di lettura delle ACL sono ancora supportate.

Come impostare le autorizzazioni ACL per un oggetto

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegliere il nome del bucket contenente l'oggetto.
3. Nell'elenco Oggetti, scegli il nome dell'oggetto per il quale si desidera impostare le autorizzazioni.
4. Seleziona Autorizzazioni.
5. In Lista di controllo degli accessi (ACL), seleziona Modifica.

   Puoi modificare le seguenti autorizzazioni ACL per l'oggetto:

   Oggetto

   - Read: consente all'assegnatario di leggere i dati dell'oggetto e i relativi metadati.

   ACL dell'oggetto

   - Read: consente all'assegnatario di leggere l'ACL dell'oggetto.
   - Write: consente all'assegnatario di scrivere l'ACL per l'oggetto interessato. Nella console S3, puoi concedere l'accesso in scrittura solo al proprietario del bucket (il tuo). Account AWS Ti consigliamo vivamente di non concedere l'accesso in scrittura ad altri utenti. Tuttavia, se devi concedere l'accesso in scrittura, puoi utilizzare gli AWS SDK o l' AWS CLI API REST.
6. È possibile gestire le autorizzazioni di accesso all'oggetto per i seguenti tipi di accesso:

   a. Accesso per il proprietario dell'oggetto

      Il proprietario si riferisce all' Utente root dell'account AWS utente IAM e non a un utente AWS Identity and Access Management IAM. Per ulteriori informazioni sull'utente root, consulta Utente root dell'account AWS nella Guida per l'utente di IAM.

      Per modificare le autorizzazioni di accesso agli oggetti del proprietario, in Accesso per il proprietario dell'oggetto, scegli Il tuo AWS account (proprietario).

Selezionare le caselle di controllo per le autorizzazioni da modificare, quindi selezionare Save (Salva).

b. Accesso per altri Account AWS

Per concedere le autorizzazioni a un AWS utente di un altro utente Account AWS, in Accesso per altri Account AWS, scegli Aggiungi account. Nel campo Inserisci un ID, inserisci l'ID canonico dell' AWS utente a cui desideri concedere le autorizzazioni relative all'oggetto. Per informazioni sulla ricerca di un ID canonico, consulta I tuoi identificatori nel. Account AWSRiferimenti generali di Amazon Web Services È possibile aggiungere fino a 99 utenti.

Selezionare le caselle di controllo relative alle autorizzazioni da concedere all'utente, quindi selezionare Save (Salva). Per visualizzare informazioni sulle autorizzazioni, scegliere le icone della Guida in linea.

c. Accesso pubblico

Per concedere al pubblico (chiunque al mondo) l'accesso all'oggetto, in Public access (Accesso pubblico) scegliere Everyone (Tutti). La concessione delle autorizzazioni di accesso pubblico consente a chiunque di accedere all'oggetto.

Selezionare le caselle di controllo per le autorizzazioni da concedere, quindi selezionare Save (Salva).

> ⚠ Warning
>
> - Prestare attenzione quando si concede al gruppo Everyone (Tutti) l'accesso anonimo agli oggetti Amazon S3. Quando si concede l'accesso a questo gruppo, qualsiasi persona al mondo può accedere all'oggetto. Se è necessario concedere l'accesso a chiunque, è vivamente consigliato farlo solo per autorizzazioni di tipo Read objects (Leggi oggetti).
> - È vivamente sconsigliato autorizzare il gruppo Everyone (Tutti) alla scrittura dell'oggetto, perché questo consentirebbe a chiunque di sovrascrivere le autorizzazioni ACL per l'oggetto.

Utilizzo degli SDK AWS

Questa sezione fornisce esempi di come configurare le autorizzazioni relative alla lista di controllo degli accessi (ACL) per i bucket e gli oggetti.

> **Important**
>
> Se il bucket utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto S3, è necessario utilizzare le policy per concedere l'accesso al bucket e agli oggetti in esso contenuti. Con l'impostazione Proprietario del bucket applicato abilitata, le richieste per impostare liste di controllo degli accessi (ACL) e aggiornare le ACL non vanno a buon fine e restituiscono il codice di errore `AccessControlListNotSupported`. Le richieste di lettura delle ACL sono ancora supportate.

Java

Questa sezione fornisce esempi di come configurare le autorizzazioni relative alla lista di controllo degli accessi (ACL) per i bucket e gli oggetti. Il primo esempio crea un bucket con un'ACL predefinita (consulta ACL predefinita), crea una lista di autorizzazioni personalizzate e poi sostituisce l'ACL predefinita con l'ACL contenente le autorizzazioni personalizzate. Il secondo esempio mostra come modificare un'ACL utilizzando il metodo `AccessControlList.grantPermission()`.

Example Creare un bucket e specificare una ACL predefinita che concede l'autorizzazione al gruppo di recapito log S3

Questo esempio crea un bucket. Nella richiesta, l'esempio specifica un'ACL predefinita che concede al Gruppo Log Delivery l'autorizzazione di scrittura dei log sul 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 Aggiornamento di un'ACL su un oggetto esistente

Questo esempio aggiorna l'ACL su un oggetto. L'esempio esegue le seguenti operazioni:

- Recupera l'ACL di un oggetto
- Elimina l'ACL rimuovendo tutte le autorizzazioni esistenti
- Aggiunge due autorizzazioni: accesso completo al proprietario e WRITE_ACP (consulta Quali autorizzazioni è possibile concedere?) per un utente identificato tramite un indirizzo email
- Salva l'ACL sull'oggetto

```
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 Creare un bucket e specificare una ACL predefinita che concede l'autorizzazione al gruppo di recapito log S3

Questo esempio C# crea un bucket. Nella richiesta, il codice specifica anche un'ACL predefinita che concede al Gruppo Log Delivery le autorizzazioni di scrittura dei log sul bucket.

Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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 Aggiornamento di un'ACL su un oggetto esistente

Questo esempio C# aggiorna l'ACL su un oggetto esistente. L'esempio esegue le seguenti operazioni:

- Recupera l'ACL di un oggetto.
- Elimina l'ACL rimuovendo tutte le autorizzazioni esistenti.
- Aggiunge due autorizzazioni: accesso completo al proprietario e WRITE_ACP per un utente identificato tramite un indirizzo email.
- Salva l'ACL inviando una richiesta `PutAcl`.

Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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());
            }
        }
    }
}
```

Utilizzo di REST API

Le API Amazon S3 permettono di impostare una lista ACL durante la creazione di un bucket o di un oggetto. Amazon S3 fornisce anche un'API per impostare una lista ACL in un bucket o un oggetto esistente. Queste API forniscono i metodi seguenti per impostare un'ACL:

- Impostazione della lista ACL tramite le intestazioni della richiesta – Quando invii una richiesta per creare una risorsa (bucket o oggetto), imposti una lista ACL utilizzando le intestazioni della richiesta. Tramite queste intestazioni, si può specificare o un'ACL predefinita oppure si possono indicare esplicitamente le concessioni (identificando assegnatario e autorizzazioni in modo esplicito).
- Impostazione della lista ACL tramite il corpo della richiesta – Quando invii una richiesta per impostare una lista ACL per una risorsa esistente, puoi impostare la lista ACL o nell'intestazione o nel corpo della richiesta.

Per informazioni sul supporto di REST API per la gestione delle liste ACL, consulta le sezioni seguenti nella Documentazione di riferimento delle API di Amazon Simple Storage Service:

- GET Bucket acl
- PUT Bucket acl
- GET Object acl
- PUT Object acl
- PUT Object
- PUT Bucket
- PUT Object - Copy
- Avvio del caricamento in più parti

> **Important**
>
> Se il bucket utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto S3, è necessario utilizzare le policy per concedere l'accesso al bucket e agli oggetti in esso contenuti. Con l'impostazione Proprietario del bucket applicato abilitata, le richieste per impostare liste di controllo degli accessi (ACL) e aggiornare le ACL non vanno a buon fine e restituiscono il codice di errore `AccessControlListNotSupported`. Le richieste di lettura delle ACL sono ancora supportate.

Intestazioni di richiesta specifiche della lista di controllo degli accessi (ACL)

È possibile utilizzare le intestazioni per concedere le autorizzazioni basate sulla lista di controllo degli accessi (ACL). Per impostazione predefinita, tutti gli oggetti sono privati. Solo il proprietario ha il controllo completo dell'accesso. Quando aggiungi un nuovo oggetto, puoi concedere autorizzazioni a singoli Account AWS o a gruppi predefiniti definiti da Amazon S3. Queste autorizzazioni vengono quindi aggiunte alla lista di controllo degli accessi (ACL) sull'oggetto. Per ulteriori informazioni, consulta Panoramica delle liste di controllo accessi (ACL).

Con questa operazione, puoi concedere le autorizzazioni di accesso utilizzando uno dei due metodi seguenti:

- ACL predefinita (**`x-amz-acl`**) - Amazon S3 supporta un set di ACL predefinite, note come ACL predefinite. Ogni ACL predefinita ha un insieme predefinito di assegnatari e autorizzazioni. Per ulteriori informazioni, consulta ACL predefinita.
- Autorizzazioni di accesso: per concedere esplicitamente le autorizzazioni di accesso a gruppi Account AWS o specifici, utilizza le seguenti intestazioni. Ogni intestazione esegue il mapping di autorizzazioni specifiche supportate da Amazon S3 in un'ACL. Per ulteriori informazioni, consulta Panoramica delle liste di controllo accessi (ACL). Nell'intestazione, specifica un elenco di assegnatari che ottengono l'autorizzazione specifica.
  - x-amz-grant-read
  - x-amz-grant-write
  - x-amz-grant-read-acp
  - x-amz-grant-write-acp
  - x-amz-grant-full-controllo

Usando il AWS CLI

Per ulteriori informazioni sulla gestione degli ACL utilizzando il AWS CLI, vedere put-bucket-aclnel AWS CLI Command Reference.

> **Important**
>
> Se il bucket utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto S3, è necessario utilizzare le policy per concedere l'accesso al bucket e agli oggetti in esso contenuti. Con l'impostazione Proprietario del bucket applicato abilitata, le richieste per impostare liste di controllo degli accessi (ACL) e aggiornare le ACL non vanno a buon fine e

restituiscono il codice di errore `AccessControlListNotSupported`. Le richieste di lettura delle ACL sono ancora supportate.

Esempi di policy per gli ACL

Puoi utilizzare le chiavi di condizione nelle policy dei bucket per controllare l'accesso ad Amazon S3.

Argomenti

- Concessione di s3: PutObject autorizzazione con una condizione che richiede al proprietario del bucket di ottenere il pieno controllo
- Concessione di s3: PutObject autorizzazione con una condizione nell'intestazione x-amz-acl

Concessione di s3: PutObject autorizzazione con una condizione che richiede al proprietario del bucket di ottenere il pieno controllo

L'operazione PUT Object permette intestazioni specifiche della lista di controllo degli accessi (ACL) che è possibile utilizzare per concedere autorizzazioni basate sulle liste ACL. Utilizzando queste chiavi, il proprietario del bucket può impostare una condizione per richiedere determinate autorizzazioni di accesso specifiche quando l'utente carica un oggetto.

Si supponga che l'Account A sia proprietario di un bucket e che l'amministratore dell'account voglia assegnare a Dave, un utente dell'Account B, le autorizzazioni per caricare oggetti. Per default, gli oggetti che carica Dave sono di proprietà dell'Account B e l'Account A non dispone di autorizzazioni su tali oggetti. Dato che il proprietario del bucket paga i conti, vuole avere le autorizzazioni complete sugli oggetti che carica Dave. L'amministratore dell'Account A può farlo assegnando l'autorizzazione `s3:PutObject` a Dave, con la condizione che la richiesta includa intestazioni specifiche della lista di controllo accessi in modo da garantire esplicitamente l'autorizzazione completa o utilizzare una lista di controllo accessi predefinita. Per ulteriori informazioni, consulta PUT Object.

Richiedi l'intestazione x-amz-full-control

È possibile richiedere l'intestazione `x-amz-full-control` nella richiesta con autorizzazione al controllo completo al proprietario del bucket. La seguente policy di bucket assegna l'autorizzazione `s3:PutObject` all'utente Dave con la condizione di utilizzare la chiave di condizione `s3:x-amz-grant-full-control` che prevede che la richiesta includa l'intestazione `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**
>
> Questo esempio riguarda l'autorizzazione tra account. Tuttavia, se Dave (che sta ottenendo l'autorizzazione) appartiene al proprietario del Account AWS bucket, questa autorizzazione condizionata non è necessaria. Questo perché l'account padre a cui Dave appartiene è proprietario degli oggetti caricati dall'utente.

Aggiunta del rifiuto esplicito

La precedente policy di bucket assegna l'autorizzazione condizionale all'utente Dave nell'Account B. Quando questa policy è attiva, per Dave è possibile ottenere la stessa autorizzazione senza alcuna condizione tramite qualche altra policy. Ad esempio, Dave può appartenere a un gruppo a cui viene assegnata l'autorizzazione `s3:PutObject` senza alcuna condizione. Per evitare questi espedienti riguardo alle autorizzazioni, è possibile scrivere una policy di accesso più rigida aggiungendo un rifiuto esplicito. In questo esempio, all'utente Dave viene esplicitamente rifiutata l'autorizzazione a eseguire caricamenti se non include le intestazioni necessarie nella richiesta che assegnano le autorizzazioni complete al proprietario del bucket. Il rifiuto esplicito sovrascrive sempre qualsiasi altra autorizzazione assegnata. Di seguito è illustrato un esempio della policy di accesso modificata con il rifiuto esplicito aggiunto.

```
{
```

```
    "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"
                }
            }
        }
    ]
}
```

Prova la politica con AWS CLI

Se ne hai due Account AWS, puoi testare la politica usando AWS Command Line Interface (AWS CLI). Allegate la policy e utilizzate le credenziali di Dave per testare l'autorizzazione usando il seguente AWS CLI `put-object` comando. Le credenziali di Dave vengono fornite aggiungendo il parametro `--profile`. L'autorizzazione al controllo completo al proprietario del bucket viene assegnata aggiungendo il parametro `--grant-full-control`. Per ulteriori informazioni sulla configurazione e l'utilizzo di AWS CLI, vedere. Sviluppo con Amazon S3 tramite la AWS CLI

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg
 --grant-full-control id="AccountA-CanonicalUserID" --profile AccountBUserProfile
```

Richiedi l' x-amz-acl intestazione

È possibile richiedere l'intestazione `x-amz-acl` con una lista di controllo degli accessi predefinita che assegna l'autorizzazione al controllo completo al proprietario del bucket. Per richiedere l'intestazione `x-amz-acl` nella richiesta, è possibile sostituire la coppia chiave-valore nel blocco `Condition` e specificare la chiave di condizione `s3:x-amz-acl` come mostrato nell'esempio seguente.

```
"Condition": {
    "StringEquals": {
        "s3:x-amz-acl": "bucket-owner-full-control"
    }
}
```

Per verificare l'autorizzazione utilizzando il AWS CLI, è necessario specificare il `--acl` parametro. AWS CLI Quindi aggiunge l'`x-amz-acl`intestazione quando invia la richiesta.

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg
 --acl "bucket-owner-full-control" --profile AccountBadmin
```

Concessione di s3: PutObject autorizzazione con una condizione nell'intestazione x-amz-acl

La seguente policy sui bucket concede l'`s3:PutObject`autorizzazione per due persone Account AWS se la richiesta include l'`x-amz-acl`intestazione che rende l'oggetto leggibile pubblicamente. Il blocco `Condition` utilizza la condizione `StringEquals` ed è dotato di una coppia chiave-valore, `"s3:x-amz-acl":["public-read"]`, per la valutazione. Nella coppia chiave-valore, la `s3:x-amz-acl` è una chiave specifica di Amazon S3, come indicato dal prefisso `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**
>
> Non tutte le condizioni hanno significato per tutte le operazioni. Ha senso, ad esempio, includere una condizione `s3:LocationConstraint` in una policy che concede l'autorizzazione `s3:CreateBucket` di Amazon S3. Non ha tuttavia senso includere questa condizione in una policy che concede l'autorizzazione `s3:GetObject`. Amazon S3 può verificare la presenza di errori semantici di questo tipo che riguardano condizioni specifiche di Amazon S3. Se tuttavia stai creando una policy per un utente o un ruolo IAM e includi una condizione di Amazon S3 che non è valida sotto il profilo semantico, non viene segnalato alcun errore perché IAM non può convalidare le condizioni di Amazon S3.

Blocco dell'accesso pubblico allo storage Amazon S3

La caratteristica di blocco dell'accesso pubblico di Amazon S3 fornisce le impostazioni per access point, bucket e account con cui è possibile gestire l'accesso pubblico alle risorse di Amazon S3. Per impostazione predefinita, nuovi bucket, access point e oggetti non consentono l'accesso pubblico. Tuttavia, gli utenti possono modificare le policy di bucket, le policy di access point o le autorizzazioni degli oggetti per consentire l'accesso pubblico. Le impostazioni di blocco dell'accesso pubblico in S3 sostituiscono le policy e le autorizzazioni, in modo da limitare l'accesso pubblico a queste risorse.

Con il blocco dell'accesso pubblico S3, gli amministratori degli account e i proprietari dei bucket possono limitare l'accesso pubblico alle risorse di Amazon S3 configurando facilmente controlli centralizzati, che vengono applicati indipendentemente dal modo in cui vengono create le risorse.

Per istruzioni sulla configurazione dell'accesso pubblico ai blocchi, consulta Configurazione del blocco dell'accesso pubblico.

Quando Amazon S3 riceve una richiesta di accesso a un bucket o a un oggetto, determina se per il bucket o l'account del proprietario del bucket è applicata un'impostazione di blocco dell'accesso pubblico. Se la richiesta è stata effettuata tramite un access point, Amazon S3 controlla anche la presenza di impostazioni di blocco dell'accesso pubblico per l'access point. Se è presente un'impostazione di blocco dell'accesso pubblico che vieta l'accesso richiesto, Amazon S3 rifiuta la richiesta.

Il blocco dell'accesso pubblico di Amazon S3 comprende quattro impostazioni. Queste impostazioni sono indipendenti e possono essere usate in qualunque combinazione. Ogni impostazione può essere applicata a un punto di accesso, a un bucket o a un intero Account AWS. Se le impostazioni di blocco dell'accesso pubblico per l'access point, il bucket o l'account sono diverse, Amazon S3 applica la combinazione più restrittiva di impostazioni.

Quando Amazon S3 valuta se un'operazione è vietata da un'impostazione di blocco dell'accesso pubblico, rifiuta qualsiasi richiesta che viola un'impostazione a livello di access point, bucket o account.

Important

L'accesso pubblico viene concesso a bucket e oggetti tramite liste di controllo accessi (ACL), policy per access point, policy di bucket o tutti. Per garantire che l'accesso pubblico sia bloccato per tutti gli access point, i bucket e gli oggetti di Amazon S3, ti consigliamo di attivare tutte e quattro le impostazioni per bloccare l'accesso pubblico per l'account. Queste impostazioni bloccano l'accesso pubblico per tutti i bucket e access point correnti e futuri. Prima di applicare queste impostazioni, verifica che le applicazioni funzionino correttamente senza accesso pubblico. Se è richiesto un certo livello di accesso pubblico ai bucket o agli oggetti, ad esempio per ospitare un sito Web statico come descritto in Hosting di un sito Web statico tramite Amazon S3, puoi personalizzare le impostazioni individuali in funzione dei casi d'uso di storage.
L'attivazione dell'accesso pubblico a blocchi aiuta a proteggere le risorse impedendo che l'accesso pubblico venga concesso tramite le politiche delle risorse o le liste di controllo degli accessi (ACL) direttamente collegate alle risorse S3. Oltre ad abilitare Block Public Access, esamina attentamente le seguenti politiche per verificare che non garantiscano l'accesso pubblico:

- Politiche basate sull'identità collegate ai AWS principali associati (ad esempio, ruoli IAM)

- Politiche basate sulle risorse collegate alle AWS risorse associate (ad esempio, chiavi (KMS)) AWS Key Management Service

> **Note**
>
> - È possibile abilitare le impostazioni di blocco dell'accesso pubblico solo per i punti di accesso, i bucket e gli Account AWS. Amazon S3 non supporta le impostazioni di blocco dell'accesso pubblico per i singoli oggetti.
> - Quando si applicano impostazioni di blocco dell'accesso pubblico a un account, le impostazioni si applicano a tutti a livello globale. Regioni AWS Le impostazioni possono non diventare effettive in tutte le regioni immediatamente o allo stesso momento, ma vengono infine propagate in tutte le regioni.

Argomenti

Impostazioni di blocco dell'accesso pubblico

Il blocco dell'accesso pubblico in S3 comprende quattro impostazioni. È possibile applicare queste impostazioni in qualsiasi combinazione a singoli access point, bucket o a interi account Account AWS. Se applichi un'impostazione a un account, l'impostazione viene applicata a tutti i bucket e gli access point di proprietà dell'account. Analogamente, se applichi un'impostazione a un bucket, questa si applica a tutti gli access point associati al bucket.

La tabella seguente contiene le impostazioni disponibili.

Nome	Descrizione
`BlockPublicAcls`	Se questa opzione è impostata su `TRUE`, produce il comportamento seguente: • Le chiamate PUT Bucket acl e PUT Object acl non riescono se la lista di controllo degli accessi (ACL) specificata è pubblica. • Le chiamate PUT Object non riescono se la richiesta include una lista di controllo accessi pubblica. • Se questa impostazione viene applicata a un account, le chiamate PUT Bucket non riescono se la richiesta include una lista di controllo accessi pubblica. Quando questa impostazione è impostata su`TRUE`, le operazioni specifica te hanno esito negativo (indipendentemente dal fatto che vengano eseguite tramite l'API REST o AWS gli SDK). AWS CLI Tuttavia, le policy e le liste di controllo accessi esistenti per bucket e oggetti non vengono modificate. Questa impostazione protegge l'ambiente dall'accesso pubblico, permetten do di controllare, perfezionare o modificare in altro modo le policy e le liste di controllo accessi per i bucket e gli oggetti. **Note** Gli access point non hanno ACL associate. Se applicata a un access point, questa impostazione funge da passthrough al bucket sottostante. Se in un access point è attivata questa impostazione, le richieste effettuate tramite l'access point si comportano come se il bucket sottostante avesse abilitato questa impostazione, indipendentemente dal fatto che il bucket abbia o meno effettiva mente abilitato questa impostazione.
`IgnorePublicAcls`	Se questa opzione è impostata su `TRUE`, Amazon S3 ignora tutte le liste ACL pubbliche in un bucket e negli oggetti contenuti. Questa impostazi one permette di bloccare in modo sicuro l'accesso pubblico concesso da

Nome	Descrizione
	liste di controllo accessi, permettendo comunque le chiamate PUT Object che includono una lista di controllo accessi pubblica (diversamente da `BlockPublicAcls` , che rifiuta le chiamate PUT Object che includono una lista di controllo accessi pubblica). L'abilitazione di questa impostazione non influisce sulla persistenza di qualsiasi lista di controllo accessi esistente e non impedisce l'impostazione di nuove liste di controllo accessi pubbliche. **Note** Gli access point non hanno ACL associate. Se applicata a un access point, questa impostazione funge da passthrough al bucket sottostante. Se in un access point è attivata questa impostazione, le richieste effettuate tramite l'access point si comportano come se il bucket sottostante avesse abilitato questa impostazione, indipendentemente dal fatto che il bucket abbia o meno effettiva mente abilitato questa impostazione.

Nome	Descrizione
`BlockPublicPolicy`	Se questa opzione è impostata su `TRUE`, Amazon S3 rifiuta le chiamate alla policy PUT dei bucket se la policy di bucket specificata permette l'accesso pubblico. Se questa opzione è impostata su `TRUE` per un bucket, Amazon S3 rifiuta le chiamate alla policy PUT dei punti di accesso per tutti i punti di accesso per lo stesso account del bucket se la policy specificata consente l'accesso pubblico. Se questa opzione è impostata su `TRUE` per un punto di accesso, Amazon S3 rifiuta le chiamate alla policy PUT dei punti di accesso e alla policy PUT dei bucket effettuata tramite il punto di accesso se la policy specificata (per il punto di accesso o il bucket sottostante) è pubblica. Puoi utilizzare questa impostazione per permettere agli utenti di gestire policy di bucket e punti di accesso impedendo loro di condividere pubblicamente il bucket o gli oggetti che contiene. L'abilitazione di questa impostazione non influisce sulle policy di access point o di bucket esistenti. **Important** Per usare questa impostazione in modo efficace, consigliamo di applicarla a livello di account. Una policy del bucket può consentire agli utenti di modificare le impostazioni di blocco dell'accesso pubblico di un bucket. Gli utenti autorizzati a modificare la policy del bucket potrebbero inserire una policy che permette loro di disabilitare le impostazioni di blocco dell'accesso pubblico per il bucket. Se questa impostazione è abilitata per l'intero account anziché per un bucket specifico, Amazon S3 blocca le policy pubbliche anche se un utente modifica la policy del bucket per disabilitare l'impostazione.

Nome	Descrizione
`RestrictPublicBuckets`	L'impostazione di questa opzione in modo da `TRUE` limitare l'accesso a un punto di accesso o a un bucket con una politica pubblica solo ai responsabili del AWS servizio e agli utenti autorizzati all'interno dell'account del proprietario del bucket e dell'account del proprietario del punto di accesso. Questa impostazione blocca tutti gli accessi tra account al punto di accesso o al bucket (ad eccezione dei responsabili del AWS servizio), pur consentendo agli utenti all'interno dell'account di gestire il punto di accesso o il bucket. L'abilitazione di questa impostazione non influisce sulle policy dell'access point o del bucket esistenti, eccetto che per il fatto che Amazon S3 blocca l'accesso pubblico e multiaccount derivato da qualsiasi policy dell'access point o del bucket pubblica, inclusa la delega non pubblica ad account specifici.

Important

- Le chiamate a GET Bucket acl e GET Object acl restituiscono sempre le autorizzazioni valide necessarie per il bucket o l'oggetto specificato. Ad esempio, supponiamo che un bucket sia associato a una lista di controllo accessi che concede l'accesso pubblico, ma che per il bucket sia anche abilitata l'impostazione `IgnorePublicAcls`. In questo caso, GET Bucket acl restituisce una lista ACL che riflette le autorizzazioni di accesso applicate da Amazon S3, anziché la lista ACL effettiva associata al bucket.
- Le impostazioni del blocco dell'accesso pubblico non modificano le policy o ACL esistenti. La rimozione di una di queste impostazioni fa sì che un bucket o un oggetto con una policy o una lista di controllo accessi pubblica torni pubblicamente accessibile.

Esecuzione di operazioni di accesso pubblico di blocco su un punto di accesso

Per eseguire operazioni di accesso pubblico a blocchi su un punto di accesso, utilizza il servizio. AWS CLI `s3control`

> **Important**
>
> Tieni presente che al momento non è possibile modificare le impostazioni di blocco dell'accesso pubblico di un access point dopo aver creato l'access point. Pertanto, l'unico modo per specificare le impostazioni di blocco dell'accesso pubblico per un access point è includerle durante la creazione dell'access point.

Significato di "pubblico"

Liste di controllo accessi

Amazon S3 considera pubblica una lista ACL di un bucket o di un oggetto se questa concede qualsiasi autorizzazione a membri dei gruppi predefiniti `AllUsers` e `AuthenticatedUsers`. Per ulteriori informazioni sui gruppi predefiniti, consulta Gruppi predefiniti di Amazon S3.

Policy di bucket

Quando valuta la policy di un bucket, Amazon S3 inizia presumendo che la policy sia pubblica. Quindi valuta la policy per determinare se si qualifica come non pubblica. Per essere considerata non pubblica, una policy di bucket deve concedere l'accesso solo a valori fissi (valori che non contengono caratteri jolly o una variabile di policy AWS Identity and Access Management) di uno o più degli elementi seguenti:

- Un AWS responsabile, un utente, un ruolo o un responsabile del servizio `aws:PrincipalOrgID` (ad es.
- Un set di Classless Inter-Domain Routing (CIDR) tramite `aws:SourceIp`. Per ulteriori informazioni sui CIDR, consulta RFC 4632 nel sito Web RFC Editor.

  > **Note**
  >
  > Le policy di bucket che concedono l'accesso in base alla chiave di condizione `aws:SourceIp` con intervalli IP molto ampi (ad esempio 0.0.0.0/1) vengono considerate "pubbliche". Sono inclusi valori superiori a /8 per IPv4 e /32 per IPv6 (esclusi gli intervalli privati RFC1918). Il blocco dell'accesso pubblico rifiuterà queste policy "pubbliche" e impedirà l'accesso multi-account ai bucket che già utilizzano queste policy "pubbliche".

- `aws:SourceArn`
- `aws:SourceVpc`

- `aws:SourceVpce`
- `aws:SourceOwner`
- `aws:SourceAccount`
- `s3:x-amz-server-side-encryption-aws-kms-key-id`
- `aws:userid`, al di fuori del modello "`AROLEID:*`"
- `s3:DataAccessPointArn`

> **Note**
>
> Se utilizzato in una policy di bucket, questo valore può contenere un carattere jolly per il nome dell'access point senza rendere pubblica la policy, purché l'ID account sia corretto. Ad esempio, consentendo l'accesso a `arn:aws:s3:us-west-2:123456789012:accesspoint/*` si consente l'accesso a qualsiasi access point associato all'account `123456789012` nella regione `us-west-2`, senza rendere pubblica la policy di bucket. Tieni presente che questo comportamento è diverso per la policy di access point. Per ulteriori informazioni, consulta Access point.

- `s3:DataAccessPointAccount`

Per ulteriori informazioni sulle policy del bucket, consulta Politiche Bucket per Amazon S3.

Example : policy di bucket pubbliche

In queste regole le policy di esempio seguenti sono considerate pubbliche.

```
{
  "Principal": "*",
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow"
 }
```

```
{
  "Principal": "*",
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow",
  "Condition": { "StringLike": {"aws:SourceVpc": "vpc-*"}}
```

```
 }
```

Queste policy possono essere modificate in non pubbliche includendo una delle chiavi di condizione elencate in precedenza, usando un valore fisso. Ad esempio, l'ultima policy indicata sopra può essere modificata in non pubblica impostando `aws:SourceVpc` su un valore fisso, come mostrato di seguito.

```
{
  "Principal": "*",
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow",
  "Condition": {"StringEquals": {"aws:SourceVpc": "vpc-91237329"}}
 }
```

Questo esempio mostra in che modo Amazon S3 valuta una policy di bucket che contiene concessioni di accesso sia pubblico sia non pubblico.

Questo esempio mostra in che modo Amazon S3 valuta una policy del bucket che contiene concessioni di accesso sia pubblico sia non pubblico.

Supponiamo che un bucket sia associato a una policy che concede l'accesso a un set di entità principali fisse. In base alle regole descritte in precedenza, questa policy non è pubblica. Di conseguenza, se abiliti l'impostazione `RestrictPublicBuckets`, la policy continua a essere valida come indicato, perché `RestrictPublicBuckets` si applica solo ai bucket associati a policy pubbliche. Tuttavia, se aggiungi un'istruzione pubblica alla policy, `RestrictPublicBuckets` ha effetto sul bucket. Consente l'accesso al bucket solo ai responsabili del AWS servizio e agli utenti autorizzati dell'account del proprietario del bucket.

Ad esempio, supponiamo che un bucket di proprietà di "Account-1" sia associato a una policy che contiene gli elementi seguenti:

1. Una dichiarazione che concede l'accesso a AWS CloudTrail (che è un servizio principale) AWS
2. Un'istruzione che concede l'accesso all'account "Account-2"
3. Un'istruzione che concede l'accesso al pubblico, ad esempio specificando `"Principal": "*"` senza `Condition` limitante

Questa policy viene qualificata come pubblica a causa della terza istruzione. Con questa politica in vigore e `RestrictPublicBuckets` abilitata, Amazon S3 consente l'accesso solo da. CloudTrail

Anche se l'istruzione 2 non è pubblica, Amazon S3 disabilita l'accesso da parte di "Account-2". Il motivo è che l'istruzione 3 rende pubblica l'intera policy, quindi viene applicata l'impostazione `RestrictPublicBuckets`. Di conseguenza, Amazon S3 disabilita l'accesso multiaccount, anche se la policy delega l'accesso a un account specifico, ovvero "Account-2". Se tuttavia rimuovi l'istruzione 3 dalla policy, questa non si qualifica più come pubblica e l'impostazione `RestrictPublicBuckets` non viene più applicata. Di conseguenza, "Account-2" riottiene l'accesso al bucket, anche se lasci abilitata l'impostazione `RestrictPublicBuckets`.

Access point

Amazon S3 valuta le impostazioni di blocco dell'accesso pubblico in modo leggermente diverso per gli access point rispetto ai bucket. Le regole applicate da Amazon S3 per determinare quando una policy di un access point è pubblica sono generalmente le stesse per gli access point e per i bucket, ad eccezione delle seguenti situazioni:

- Un access point con un'origine di rete VPC è sempre considerato non pubblico, indipendentemente dal contenuto della policy di access point.
- Una policy di access point che concede l'accesso a un set di access point utilizzando `s3:DataAccessPointArn` è considerata pubblica. Tieni presente che questo comportamento è diverso rispetto alle policy di bucket. Ad esempio, una policy di bucket che concede l'accesso ai valori di `s3:DataAccessPointArn` che corrispondono a `arn:aws:s3:us-west-2:123456789012:accesspoint/*` è considerata pubblica. Tuttavia, la stessa istruzione in una policy di access point renderebbe pubblico l'access point.

Utilizzo di IAM Access Analyzer per S3 per esaminare i bucket pubblici

È possibile utilizzare IAM Access Analyzer per S3 per esaminare i bucket con ACL bucket, policy di bucket o policy del punto di accesso che concedono l'accesso pubblico. IAM Access Analyzer for S3 ti avvisa della presenza di bucket configurati per consentire l'accesso a chiunque sia connesso a Internet o altro Account AWS, anche Account AWS all'esterno dell'organizzazione. Per ogni bucket pubblico o condiviso, vengono visualizzati risultati che riportano l'origine e il livello di accesso pubblico o condiviso.

In IAM Access Analyzer per S3, è possibile bloccare tutti gli accessi pubblici a un bucket con un solo clic. Inoltre, puoi eseguire il drill-down nelle impostazioni relative alle autorizzazioni a livello di bucket per configurare i livelli di accesso granulari. Per casi d'uso specifici e verificati che richiedono l'accesso pubblico o condiviso, puoi confermare e registrare l'intenzione del bucket di rimanere pubblico o condiviso archiviando i risultati per il bucket.

In rari casi, IAM Access Analyzer per S3 può segnalare l'assenza di risultati per un bucket che una valutazione del blocco dell'accesso pubblico Amazon S3 segnala come pubblico. Ciò accade perché il blocco dell'accesso pubblico di Amazon S3 esamina le policy per le operazioni correnti ed eventuali operazioni potenziali che potrebbero venire aggiunte in futuro, facendo sì che un bucket diventi pubblico. D'altra parte, IAM Access Analyzer per S3 analizza solo le azioni correnti specificate per il servizio Amazon S3 nella valutazione dello stato di accesso.

Per ulteriori informazioni su IAM Access Analyzer per S3, consultare Revisione dell'accesso al bucket tramite IAM Access Analyzer per S3.

Autorizzazioni

Per utilizzare le caratteristiche di blocco dell'accesso pubblico di Amazon S3, sono necessarie le autorizzazioni seguenti.

Operazione	Autorizzazioni richieste
Operazione GET per lo stato della policy del bucket	`s3:GetBucketPolicyStatus`
Operazione GET per le impostazioni di blocco dell'accesso pubblico del bucket	`s3:GetBucketPublicAccessBlock`
Operazione PUT per le impostazioni di blocco dell'accesso pubblico del bucket	`s3:PutBucketPublicAccessBlock`
Operazione DELETE per le impostazioni di blocco dell'accesso pubblico del bucket	`s3:PutBucketPublicAccessBlock`
Operazione GET per le impostazioni di blocco dell'accesso pubblico dell'account	`s3:GetAccountPublicAccessBlock`
Operazione PUT per le impostazioni di blocco dell'accesso pubblico dell'account	`s3:PutAccountPublicAccessBlock`
Operazione DELETE per le impostazioni di blocco dell'accesso pubblico dell'account	`s3:PutAccountPublicAccessBlock`
Operazione PUT per le impostazioni di blocco dell'accesso pubblico dell'access point	`s3:CreateAccessPoint`

> **Note**
>
> Per le operazioni DELETE sono necessarie le stesse autorizzazioni necessarie per le operazioni PUT. Non esistono autorizzazioni separate per le operazioni DELETE.

Configurazione del blocco dell'accesso pubblico

Per ulteriori informazioni sulla configurazione dell'accesso pubblico a blocchi per i tuoi bucket Amazon S3 Account AWS e per i tuoi bucket Amazon S3, consulta i seguenti argomenti.

- Configurazione delle impostazioni di blocco dell'accesso pubblico per l'account
- Configurazione delle impostazioni di blocco dell'accesso pubblico per i bucket S3

Configurazione delle impostazioni di blocco dell'accesso pubblico per l'account

Il blocco dell'accesso pubblico di Amazon S3 fornisce le impostazioni per punti di accesso, bucket e account con cui è possibile gestire l'accesso pubblico alle risorse di Amazon S3. Per impostazione predefinita, nuovi bucket, punti di accesso e oggetti non consentono l'accesso pubblico.

Per ulteriori informazioni, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

> **Note**
>
> Le impostazioni a livello di account sostituiscono le impostazioni sui singoli oggetti. Se si configura l'account per bloccare l'accesso pubblico, le eventuali impostazioni di accesso pubblico effettuate su singoli oggetti all'interno dell'account verranno sovrascritte.

Puoi utilizzare la console S3 AWS CLI, gli AWS SDK e l'API REST per configurare le impostazioni di accesso pubblico a blocchi per tutti i bucket del tuo account. Per ulteriori informazioni, consulta le sezioni seguenti.

Per configurare le impostazioni di accesso pubblico di blocco per i bucket, consulta Configurazione delle impostazioni di blocco dell'accesso pubblico per i bucket S3. Per ulteriori informazioni sui punti di accesso, consulta Esecuzione di operazioni di accesso pubblico di blocco su un punto di accesso.

Utilizzo della console S3

Il blocco dell'accesso pubblico di Amazon S3 impedisce l'applicazione di qualsiasi impostazione che consente l'accesso pubblico ai dati all'interno di bucket S3. In questa sezione viene descritto come modificare le impostazioni di blocco dell'accesso pubblico per tutti i bucket S3 nell' Account AWS. Per ulteriori informazioni sul blocco dell'accesso pubblico, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

Per modificare le impostazioni di accesso pubblico a blocchi per tutti i bucket S3 in un Account AWS

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Scegli Impostazioni account per blocco dell'accesso pubblico.
3. Scegli Modifica per modificare le impostazioni di blocco dell'accesso pubblico per tutti i bucket nell' Account AWS.
4. Scegliere le impostazione da modificare, quindi selezionare Save changes (Salva modifiche).
5. Quando viene richiesta la conferma, immettere **`confirm`**. Quindi scegliere Confirm (Conferma) per salvare le modifiche.

Usando il AWS CLI

È possibile utilizzare il blocco dell'accesso pubblico di Amazon S3 tramite la AWS CLI. Per ulteriori informazioni sulla configurazione e l'utilizzo di AWS CLI, vedi Cos'è il AWS Command Line Interface?

Account

Per eseguire operazioni di blocco dell'accesso pubblico su un account, usa il servizio `s3control` di AWS CLI . Le operazioni a livello di account che usano questo servizio sono:

- PUT PublicAccessBlock (per un account)
- GET PublicAccessBlock (per un account)
- DELETE PublicAccessBlock (per un account)

Per ulteriori informazioni ed esempi, vedere put-public-access-blocknella Guida di AWS CLI riferimento.

Utilizzo degli AWS SDK

Java

Gli esempi seguenti mostrano come utilizzare Amazon S3 Block Public Access con AWS SDK for Java per inserire una configurazione di blocco di accesso pubblico su un account 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

Questo esempio si applica solo alle operazioni a livello di account, che usano la classe client `AWSS3Control`. Per le operazioni a livello di bucket, consulta l'esempio precedente.

Other SDKs

Per informazioni sull'utilizzo degli altri AWS SDK, consulta. Sviluppo con Amazon S3 utilizzando gli SDK AWS

Utilizzo di REST API

Per informazioni sull'utilizzo del blocco dell'accesso pubblico di Amazon S3 tramite le REST API, consulta i seguenti argomenti nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

- Operazioni a livello di account
  - METTERE PublicAccessBlock
  - OTTIENI PublicAccessBlock
  - ELIMINA PublicAccessBlock

Configurazione delle impostazioni di blocco dell'accesso pubblico per i bucket S3

Il blocco dell'accesso pubblico di Amazon S3 fornisce le impostazioni per punti di accesso, bucket e account con cui è possibile gestire l'accesso pubblico alle risorse di Amazon S3. Per impostazione predefinita, nuovi bucket, punti di accesso e oggetti non consentono l'accesso pubblico.

Per ulteriori informazioni, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

Puoi utilizzare la console S3 AWS CLI, gli AWS SDK e l'API REST per concedere l'accesso pubblico a uno o più bucket. È anche possibile bloccare l'accesso pubblico a bucket che sono già pubblici. Per ulteriori informazioni, consulta le sezioni seguenti.

Per configurare le impostazioni Blocco dell'accesso pubblico per ogni bucket dell'account, consultare Configurazione delle impostazioni di blocco dell'accesso pubblico per l'account. Per informazioni sulla configurazione dell'accesso pubblico a blocchi per i punti di accesso, consulta Esecuzione di operazioni di accesso pubblico di blocco su un punto di accesso.

Utilizzo della console S3

Il blocco dell'accesso pubblico Amazon S3 impedisce l'applicazione di qualsiasi impostazione che consente l'accesso pubblico ai dati all'interno dei bucket S3. In questa sezione viene descritto come modificare le impostazioni di blocco dell'accesso pubblico per uno o più bucket S3. Per informazioni sul blocco dell'accesso pubblico utilizzando gli AWS CLI AWS SDK e le API REST di Amazon S3, consulta. Blocco dell'accesso pubblico allo storage Amazon S3

Puoi vedere se il tuo bucket è accessibile pubblicamente nell'elenco Bucket. Nella colonna Accesso, Amazon S3 etichetta le autorizzazioni per un bucket nel modo seguente:

- Public (Pubblico): tutti hanno accesso a uno o più dei seguenti elementi: oggetti elenco, oggetti in scrittura, autorizzazioni di lettura e scrittura.
- Objects can be public (Gli oggetti possono essere pubblici): il bucket non è pubblico, ma chiunque disponga delle autorizzazioni appropriate può concedere l'accesso pubblico agli oggetti.
- Buckets and objects not public (Bucket e oggetti non pubblici): il bucket e gli oggetti non hanno accesso pubblico.

- Solo utenti autorizzati di questo account: l'accesso è limitato agli utenti e ai ruoli IAM relativi a questo account e ai responsabili del AWS servizio, poiché esiste una politica che garantisce l'accesso pubblico.

È anche possibile filtrare le ricerche di bucket in base al tipo di accesso. Scegliere un tipo di accesso dall'elenco a discesa disponibile accanto alla barra Search for buckets (Cerca bucket).

Se viene visualizzato un `Error` quando si elencano i bucket e le relative impostazioni di accesso pubblico, si potrebbe non disporre delle autorizzazioni richieste. Assicurarsi di disporre delle seguenti autorizzazioni aggiunte alla policy utente o del ruolo:

```
s3:GetAccountPublicAccessBlock
s3:GetBucketPublicAccessBlock
s3:GetBucketPolicyStatus
s3:GetBucketLocation
s3:GetBucketAcl
s3:ListAccessPoints
s3:ListAllMyBuckets
```

In alcuni rari casi, le richieste possono anche non riuscire a causa di un'interruzione della Regione AWS .

Per modificare le impostazioni di blocco dell'accesso pubblico Amazon S3 per un singolo bucket S3

Segui questa procedura se è necessario modificare le impostazioni di accesso pubblico per un singolo bucket S3.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket name (Nome bucket), scegliere il nome del bucket desiderato.
3. Seleziona Autorizzazioni.
4. Scegliere Edit (Modifica) per modificare le impostazioni di accesso pubblico per il bucket. Per maggiori informazioni sulle quattro impostazioni di blocco dell'accesso pubblico di Amazon S3, consulta Impostazioni di blocco dell'accesso pubblico.
5. Scegliere l'impostazione da modificare, quindi selezionare Save (Salva).
6. Quando viene richiesta la conferma, immettere **`confirm`**. Quindi scegliere Confirm (Conferma) per salvare le modifiche.

Puoi modificare le impostazioni di blocco dell'accesso pubblico Amazon S3 al momento della creazione di un bucket. Per ulteriori informazioni, consulta Creazione di un bucket.

Usando il AWS CLI

Per bloccare l'accesso pubblico su un bucket o eliminare il blocco di accesso pubblico, utilizza il AWS CLI servizio `s3api`. Le operazioni a livello di bucket che usano questo servizio sono:

- PUT PublicAccessBlock (per un secchio)
- GET PublicAccessBlock (per un secchio)
- DELETE PublicAccessBlock (per un secchio)
- OTTIENI BucketPolicyStatus

Per ulteriori informazioni ed esempi, consulta la put-public-access-blocksezione AWS CLI Reference.

Utilizzo degli 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**
>
> Questo esempio si applica solo alle operazioni a livello di bucket, che usano la classe client `AmazonS3`. Per le operazioni a livello di account, consulta l'esempio seguente.

Other SDKs

Per informazioni sull'utilizzo degli altri AWS SDK, consulta. Sviluppo con Amazon S3 utilizzando gli SDK AWS

Utilizzo di REST API

Per informazioni sull'utilizzo del blocco dell'accesso pubblico di Amazon S3 tramite le REST API, consulta i seguenti argomenti nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

- Operazioni a livello di bucket
  - METTERE PublicAccessBlock
  - OTTIENI PublicAccessBlock
  - ELIMINA PublicAccessBlock
  - OTTIENI BucketPolicyStatus

Revisione dell'accesso al bucket tramite IAM Access Analyzer per S3

IAM Access Analyzer for S3 ti avvisa della presenza di bucket S3 configurati per consentire l'accesso a chiunque su Internet o altro Account AWS, anche Account AWS all'esterno della tua organizzazione. Per ogni bucket pubblico o condiviso, vengono visualizzati risultati per l'origine e il livello di accesso pubblico o condiviso. Ad esempio, IAM Access Analyzer per S3 potrebbe mostrare che un bucket dispone di accesso in lettura o scrittura fornito tramite una lista di controllo degli accessi (ACL) del bucket, una policy del bucket, una policy del punto di accesso multi-regione o una policy del punto di accesso. Con questi risultati puoi intraprendere azioni correttive immediate e precise per ripristinare l'accesso del bucket desiderato.

Durante la revisione di un bucket a rischio in IAM Access Analyzer per S3, è possibile bloccare tutti gli accessi pubblici al bucket con un solo clic. Ti consigliamo di bloccare tutti gli accessi ai bucket, a meno che non sia necessario l'accesso pubblico per supportare un caso d'uso specifico. Prima di bloccare tutti gli accessi pubblici, assicurati che le applicazioni continuino a funzionare correttamente senza accesso pubblico. Per ulteriori informazioni, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

Inoltre, puoi eseguire il drill-down nelle impostazioni relative alle autorizzazioni a livello di bucket per configurare i livelli di accesso granulari. Per casi d'uso specifici e verificati che richiedono l'accesso

pubblico, ad esempio host di siti Web, download pubblici, condivisione tra account, puoi confermare e registrare l'intenzione del bucket di rimanere pubblico o condiviso archiviando i risultati per il bucket. Puoi consultare e modificare le configurazioni relative al bucket in qualsiasi momento. Inoltre, puoi scaricare i risultati come report CSV per scopi di audit.

IAM Access Analyzer per S3 è disponibile senza costi aggiuntivi nella console di Amazon S3. IAM Access Analyzer per S3 è basato su AWS Identity and Access Management (IAM) IAM Access Analyzer. Per utilizzare IAM Access Analyzer for S3 nella console Amazon S3, devi visitare la console IAM e abilitare IAM Access Analyzer per regione.

Per ulteriori informazioni su IAM Access Analyzer, consulta Cos'è IAM Access Analyzer? nella Guida per l'utente di IAM. Per ulteriori informazioni su IAM Access Analyzer per S3, rivedi le sezioni seguenti.

Important

- IAM Access Analyzer per S3 richiede un analizzatore a livello di account. Per utilizzare IAM Access Analyzer for S3, devi visitare IAM Access Analyzer e creare un analizzatore che abbia un account come zona di fiducia. Per ulteriori informazioni, consultare Abilitazione di IAM Access Analyzer nella Guida per l'utente di IAM.
- IAM Access Analyzer per S3 non analizza la policy dei punti di accesso associata ai punti di accesso multi-account. Questo comportamento si verifica perché il punto di accesso e la relativa policy sono al di fuori della zona di attendibilità, ovvero l'account. I bucket che delegano l'accesso a un punto di accesso multi-account sono elencati in Buckets with public access (Bucket con accesso pubblico) se non hai applicato l'impostazione `RestrictPublicBuckets` di Blocco dell'accesso pubblico Amazon S3 al bucket o all'account. Quando applichi l'impostazione di `RestrictPublicBuckets` blocco dell'accesso pubblico, il bucket viene riportato in Bucket con accesso da altri, inclusi quelli di terze parti. Account AWS Account AWS
- Quando una policy del bucket o un'ACL bucket viene aggiunta o modificata, IAM Access Analyzer genera e aggiorna i risultati in base alla modifica entro 30 minuti. I risultati relativi alle impostazioni di Blocco dell'accesso pubblico Amazon S3 a livello di account potrebbero non essere generati o aggiornati per un massimo di 6 ore dopo la modifica delle impostazioni. I risultati relativi ai punti di accesso multi-regione potrebbero non essere generati o aggiornati per un massimo di sei ore dopo la creazione o l'eliminazione del punto di accesso multi-regione o della modifica della policy.

Argomenti

Quali informazioni sono fornite da IAM Access Analyzer per S3?

IAM Access Analyzer per S3 fornisce risultati per i bucket a cui è possibile accedere al di fuori di Account AWS. I bucket elencati sotto Buckets with public access (Bucket con accesso pubblico) sono accessibili da chiunque su Internet. Se IAM Access Analyzer per S3 identifica i bucket pubblici, nella parte superiore della pagina viene visualizzato un avviso che indica il numero di bucket pubblici nella regione. I bucket elencati nella sezione Bucket con accesso da altri Account AWS , inclusi quelli di terze parti, Account AWS vengono condivisi in modo condizionale con altri Account AWS, compresi gli account esterni all'organizzazione.

Per ogni bucket, IAM Access Analyzer for S3 fornisce le seguenti informazioni:

- Nome bucket
- Rilevato da Access Analyzer - Quando IAM Access Analyzer per S3 ha rilevato l'accesso al bucket pubblico o condiviso.
- Condiviso tramite: come il bucket viene condiviso, ovvero tramite una policy di bucket, una ACL di bucket, una policy del punto di accesso multi-regione o una policy del punto di accesso. I punti di accesso multi-regione e i punti di accesso multi-account sono riportati sotto i punti di accesso. Un bucket può essere condiviso tramite policy e ACL. Per trovare e rivedere l'origine dell'accesso al bucket, puoi utilizzare le informazioni contenute in questa colonna come punto di partenza per intraprendere azioni correttive immediate e precise.
- Status (Stato) - Lo stato del rilevamento del bucket. IAM Access Analyzer per S3 visualizza i risultati per tutti i bucket pubblici e condivisi.
  - Active (Attivo)- Il risultato non è stato esaminato.

  - Archived (Archiviato) - Il risultato è stato esaminato e confermato come previsto.
  - Tutti - Tutti i risultati relativi ai bucket che sono pubblici o condivisi con altri Account AWS, anche Account AWS al di fuori dell'organizzazione.
- Access level (Livello di accesso) - Autorizzazioni di accesso concesse per il bucket:
  - List (Elenco) - Elencare le risorse.
  - Read (Lettura) - Leggere ma non modificare gli attributi e i contenuti delle risorse.
  - Write (Scrittura) - Creare, eliminare o modificare le risorse.
  - Permissions (Autorizzazioni) - Concedere o modificare le autorizzazioni a livello di risorsa.
  - Tagging (Tag) - Aggiornare i tag associati alla risorsa.

Abilitazione di IAM Access Analyzer per S3

Per utilizzare IAM Access Analyzer per S3, è necessario completare i seguenti passaggi prerequisiti.

1. Concedere le autorizzazioni richieste.

   Per ulteriori informazioni, consultare Autorizzazioni necessarie per utilizzare IAM Access Analyzer nella Guida per l'utente di IAM.
2. Visita IAM per creare un analizzatore a livello di account per ogni regione in cui desideri utilizzare IAM Access Analyzer.

   IAM Access Analyzer per S3 richiede un analizzatore a livello di account. Per utilizzare IAM Access Analyzer per S3, è necessario creare un analizzatore con un account come zona di attendibilità. Per ulteriori informazioni, consultare Abilitazione di IAM Access Analyzer nella Guida per l'utente di IAM.

Blocco di tutti gli accessi pubblici

Per bloccare tutti gli accessi a un bucket con un solo clic, puoi utilizzare il pulsante Blocca tutti gli accessi pubblici in IAM Access Analyzer per S3. Quando blocchi tutti gli accessi pubblici a un bucket, non viene concesso alcun accesso pubblico. Ti consigliamo di bloccare tutti gli accessi pubblici ai bucket, a meno che non sia necessario l'accesso pubblico per supportare un caso d'uso specifico e verificato. Prima di bloccare tutti gli accessi pubblici, assicurati che le applicazioni continuino a funzionare correttamente senza accesso pubblico.

Se non desideri bloccare tutti gli accessi pubblici al bucket, puoi modificare le impostazioni di blocco dell'accesso pubblico sulla console di Amazon S3 per configurare livelli granulari di accesso ai bucket. Per ulteriori informazioni, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

In rari casi, IAM Access Analyzer per S3 può segnalare l'assenza di risultati per un bucket che una valutazione del blocco dell'accesso pubblico Amazon S3 segnala come pubblico. Ciò accade perché il blocco dell'accesso pubblico di Amazon S3 esamina le policy per le operazioni correnti ed eventuali operazioni potenziali che potrebbero venire aggiunte in futuro, facendo sì che un bucket diventi pubblico. D'altra parte, IAM Access Analyzer per S3 analizza solo le azioni correnti specificate per il servizio Amazon S3 nella valutazione dello stato di accesso.

Per bloccare tutti gli accessi pubblici a un bucket utilizzando IAM Access Analyzer per S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, in Dashboards (Pannelli di controllo), scegliere Access analyzer for S3 (Access Analyzer per S3).
3. In IAM Access Analyzer per S3, scegli un bucket.
4. Scegliere Block all public access (Blocca tutti gli accessi pubblici).
5. Per confermare l'intenzione di bloccare tutti gli accessi pubblici al bucket, in Block all public access (bucket settings) (Blocca tutti gli accessi pubblici (impostazioni bucket)), immettere **`confirm`**.

   Amazon S3 blocca tutti gli accessi pubblici al bucket. Lo stato del risultato del bucket viene aggiornato in risolto e il bucket scompare dall'elenco di IAM Access Analyzer per S3. Se desideri esaminare i bucket risolti, apri IAM Access Analyzer sulla console IAM.

Revisione e modifica dell'accesso al bucket

Se non intendi concedere l'accesso al pubblico o ad altri Account AWS, compresi gli account esterni alla tua organizzazione, puoi modificare l'ACL del bucket, la policy del bucket, la politica del punto di accesso multiregionale o la politica del punto di accesso per rimuovere l'accesso al bucket. La colonna Shared through (Condiviso tramite) mostra tutte le origini dell'accesso al bucket: policy di bucket, ACL di bucket e/o policy del punto di accesso. I punti di accesso multi-regione e i punti di accesso multi-account sono riportati sotto i punti di accesso.

Per esaminare e modificare una policy di bucket, una ACL, una policy del punto di accesso multi-regione o del punto di accesso di un bucket

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione scegliere Access analyzer for S3 (Access Analyzer per S3).
3. Per verificare se l'accesso pubblico o l'accesso condiviso è concesso tramite una policy di bucket, una ACL di bucket, una policy del punto di accesso multi-regione o una policy del punto di accesso, cerca nella colonna Shared through (Condiviso tramite).
4. In Buckets (Bucket) scegli il nome del bucket con la policy di bucket, l'ACL di bucket, la policy del punto di accesso multi-regione o la policy del punto di accesso che desideri modificare o esaminare.
5. Se si desidera modificare o visualizzare una ACL di bucket:
   a. Seleziona Autorizzazioni.
   b. Scegliere Access Control List (Lista di controllo accessi).
   c. Esaminare l'ACL di bucket e apportare le modifiche necessarie.

      Per ulteriori informazioni, consulta Configurazione delle ACL.
6. Se si desidera modificare o rivedere una policy di bucket:
   a. Seleziona Autorizzazioni.
   b. Scegli Bucket Policy (Policy del bucket).
   c. Esaminare o modificare la policy di bucket come richiesto.

      Per ulteriori informazioni, consulta Aggiunta di una policy di bucket utilizzando la console di Amazon S3.
7. Se desideri esaminare o modificare una policy del punto di accesso multi-regione:
   a. Scegli Multi-Region Access Point (Punto di accesso multi-regione).
   b. Scegli il nome del punto di accesso multi-regione.
   c. Esamina o modifica la policy del punto di accesso multi-regione come necessario.

      Per ulteriori informazioni, consulta Autorizzazioni.
8. Se si desidera rivedere o modificare una policy del punto di accesso:
   a. Scegliere Access points (Access point).

b. Scegliere il nome del punto di accesso.

c. Esaminare o modificare l'accesso in base alle esigenze.

Per ulteriori informazioni, consulta Gestione e utilizzo degli Punti di accesso Amazon S3 nella console di Amazon S3.

Se si modifica o si rimuove una ACL di bucket, una policy di bucket o una policy del punto di accesso per rimuovere l'accesso pubblico o condiviso, lo stato dei risultati del bucket viene aggiornato in resolved (risolto). I risultati risolti del bucket scompaiono dall'elenco di IAM Access Analyzer per S3, ma puoi visualizzarli in IAM Access Analyzer.

Archiviazione dei risultati del bucket

Se un bucket consente l'accesso al pubblico o ad altri utenti Account AWS, inclusi account esterni all'organizzazione, per supportare un caso d'uso specifico (ad esempio, un sito Web statico, download pubblici o condivisione tra account), puoi archiviare i risultati relativi al bucket. Quando archivi i risultati del bucket, confermi e registri l'intenzione che il bucket rimanga pubblico o condiviso. I risultati del bucket archiviati rimangono nell'elenco di IAM Access Analyzer per S3 in modo da sapere sempre quali bucket sono pubblici o condivisi.

Per archiviare i risultati del bucket in IAM Access Analyzer per S3

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione scegliere Access analyzer for S3 (Access Analyzer per S3).
3. In IAM Access Analyzer per S3, scegli un bucket attivo.
4. Per confermare l'intenzione di consentire l'accesso a questo bucket da parte del pubblico o di altri utenti Account AWS, inclusi gli account esterni all'organizzazione, scegli Archivia.
5. Immettere **`confirm`** e scegliere Archive (Archivia).

Attivazione di un risultato di bucket archiviato

Dopo aver archiviato i risultati, è sempre possibile esaminarli e modificarne lo stato attivo, indicando che il bucket richiede un'altra revisione.

Per attivare un risultato di bucket archiviato in IAM Access Analyzer per S3

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.

2. Nel riquadro di navigazione scegliere Access analyzer for S3 (Access Analyzer per S3).
3. Scegliere i risultati del bucket archiviati.
4. Scegliere Mark as active (Contrassegna come attivo).

Visualizzazione dei dettagli del risultato

Se hai bisogno di visualizzare ulteriori informazioni su un bucket, puoi aprirlo trovando i dettagli in IAM Access Analyzer sulla console IAM.

Per visualizzare i dettagli dei risultati in IAM Access Analyzer per S3

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione scegliere Access analyzer for S3 (Access Analyzer per S3).
3. In IAM Access Analyzer per S3, scegli un bucket.
4. Seleziona View Details (Visualizza dettagli).

   I dettagli del risultato si trovano in IAM Access Analyzer sulla console IAM.

Download di un report IAM Access Analyzer per S3

Puoi scaricare i risultati del bucket come report CSV che è possibile utilizzare per scopi di audit. Il report include le stesse informazioni visualizzate in IAM Access Analyzer per S3 sulla console di Amazon S3.

Per scaricare un report

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra scegliere Access analyzer for S3 (Access Analyzer per S3).
3. Nel filtro Region (Regione) scegliere Region (Regione).

   IAM Access Analyzer per S3 viene aggiornato per mostrare i bucket per la regione scelta.
4. Scegliere Download report (Scarica report).

   Un report CSV viene generato e salvato sul computer.

Verifica della proprietà del bucket con condizione del proprietario del bucket

La condizione di proprietario dei bucket Amazon S3 garantisce che i bucket utilizzati nelle operazioni S3 appartengano a quelli previsti. Account AWS

La maggior parte delle operazioni S3 legge da o scrive in bucket S3 specifici. Queste operazioni includono il caricamento, la copia e il download di oggetti, il recupero o la modifica delle configurazioni dei bucket e il recupero o la modifica delle configurazioni degli oggetti. Quando esegui queste operazioni, specifichi il bucket da utilizzare includendo il suo nome nella richiesta. Ad esempio, per recuperare un oggetto da S3, effettui una richiesta che specifica il nome di un bucket e la chiave oggetto da recuperare da quel bucket.

Poiché Amazon S3 identifica i bucket in base ai loro nomi, un'applicazione che utilizza un nome bucket non corretto in una richiesta potrebbe accidentalmente eseguire operazioni su un bucket diverso da quello previsto. Per evitare interazioni del bucket involontarie in situazioni come questa, puoi utilizzare la condizione proprietario del bucket. La condizione proprietario del bucket consente di verificare che il bucket di destinazione sia di proprietà dell' Account AWS previsto, fornendo un ulteriore livello di garanzia sul fatto che le operazioni S3 avranno gli effetti desiderati.

Argomenti

Quando utilizzare la condizione proprietario del bucket

È consigliabile utilizzare la condizione proprietario del bucket ogni volta che esegui un'operazione S3 supportata e conosci l'ID account del proprietario del bucket previsto. La condizione proprietario del bucket è disponibile per tutte le operazioni oggetto S3 e la maggior parte delle operazioni bucket S3. Per un elenco delle operazioni S3 che non supportano la condizione proprietario del bucket, consulta Restrizioni e limitazioni.

Per scoprire i vantaggi derivanti dall'utilizzo della condizione di proprietario del bucket, considera il seguente scenario che coinvolge il cliente Bea: AWS

1. Bea sviluppa un'applicazione che utilizza Amazon S3. Durante lo sviluppo, Bea utilizza i suoi test solo Account AWS per creare un bucket denominato `bea-data-test` e configura la sua applicazione per effettuare richieste a. `bea-data-test`
2. Bea distribuisce la sua applicazione, ma dimentica di riconfigurarla affinché utilizzi un bucket nel suo Account AWS di produzione.
3. In produzione, l'applicazione di Bea effettua richieste a `bea-data-test`, che hanno esito positivo. In questo modo i dati di produzione vengono scritti nel bucket nell'account di test di Bea.

Bea può prevenire situazioni come questa utilizzando la condizione proprietario del bucket. Con la condizione di proprietario del bucket, Bea può includere l' Account AWS ID del proprietario del bucket previsto nelle sue richieste. Amazon S3 controlla quindi l'ID account del proprietario del bucket prima di elaborare ogni richiesta. Se il proprietario del bucket effettivo non corrisponde al proprietario del bucket previsto, la richiesta ha esito negativo.

Se Bea utilizza la condizione proprietario del bucket, lo scenario descritto in precedenza non comporta la scrittura accidentale dell'applicazione di Bea in un bucket di test. Invece, le richieste effettuate dall'applicazione nella fase 3 avranno esito negativo con un messaggio di errore `Access Denied`. Utilizzando la condizione proprietario del bucket, Bea aiuta a eliminare il rischio di interagire accidentalmente con i bucket nell' Account AWS sbagliato.

Verifica del proprietario del bucket

Per utilizzare la condizione proprietario del bucket, includi nella richiesta un parametro che specifichi il proprietario del bucket previsto. La maggior parte delle operazioni S3 coinvolge solo un singolo bucket e richiede solo questo singolo parametro per utilizzare la condizione proprietario del bucket. Per le operazioni `CopyObject`, questo primo parametro specifica il proprietario previsto del bucket di destinazione e viene incluso un secondo parametro per specificare il proprietario previsto del bucket di origine.

Quando effettui una richiesta che include un parametro della condizione proprietario del bucket, S3 controlla l'ID account del proprietario del bucket rispetto al parametro specificato prima di elaborare la richiesta. Se il parametro corrisponde all'ID account del proprietario del bucket, S3 elabora la richiesta. Se il parametro non corrisponde all'ID account del proprietario del bucket, la richiesta ha esito negativo con un messaggio di errore `Access Denied`.

Puoi usare la condizione del proprietario del bucket con AWS Command Line Interface (AWS CLI), gli AWS SDK e le API REST di Amazon S3. Quando utilizzi la condizione di proprietario del bucket con AWS CLI le API REST di Amazon S3, usa i seguenti nomi di parametri.

Metodo di accesso	Parametro per operazioni di non copia	Parametro origine operazione di copia	Parametro destinazione operazione di copia
AWS CLI	`--expected-bucket-owner`	`--expected-source-bucket-owner`	`--expected-bucket-owner`
REST API di Amazon S3	`x-amz-expected-bucket-owner` Intestazione	`x-amz-source-expected-bucket-owner` Intestazione	`x-amz-expected-bucket-owner` Intestazione

I nomi dei parametri necessari per utilizzare la condizione proprietario del bucket con gli SDK AWS variano a seconda della lingua. Per determinare i parametri richiesti, consulta la documentazione SDK relativa alla lingua desiderata. È possibile trovare la documentazione SDK in Strumenti per creare in AWS.

Esempi

Gli esempi seguenti mostrano come implementare la condizione di proprietario del bucket in Amazon S3 utilizzando o AWS CLI il. AWS SDK for Java 2.x

Example

Esempio: caricare un oggetto

Nell'esempio seguente viene mostrato il caricamento di un oggetto nel bucket S3 *example-s3-bucket1*, utilizzando la condizione di proprietario del bucket per assicurarsi che *example-s3-bucket1* sia di proprietà dell' Account AWS 111122223333.

AWS CLI

```
aws s3api put-object \
                --bucket example-s3-bucket1 --key exampleobject --
body example_file.txt \
                --expected-bucket-owner 111122223333
```

AWS SDK for Java 2.x

```
public void putObjectExample() {
    S3Client s3Client = S3Client.create();;
    PutObjectRequest request = PutObjectRequest.builder()
            .bucket("example-s3-bucket1")
            .key("exampleobject")
            .expectedBucketOwner("111122223333")
            .build();
    Path path = Paths.get("example_file.txt");
    s3Client.putObject(request, path);
}
```

Example

Esempio: copiare un oggetto

Nell'esempio seguente viene mostrata la copia di un oggetto `object1` dal bucket S3 *example-s3-bucket1* nel bucket S3 *example-s3-bucket2*. Utilizza la condizione proprietario del bucket per garantire che i bucket sono di proprietà degli account previsti secondo la tabella seguente.

Bucket	Proprietario previsto
example-s3-bucket1	111122223333
example-s3-bucket2	444455556666

AWS CLI

```
aws s3api copy-object --copy-source example-s3-bucket1/object1 \
                          --bucket example-s3-bucket2 --key object1copy \
                          --expected-source-bucket-owner 111122223333 --expected-
bucket-owner 444455556666
```

AWS SDK for Java 2.x

```
public void copyObjectExample() {
        S3Client s3Client = S3Client.create();
        CopyObjectRequest request = CopyObjectRequest.builder()
```

```
                .copySource("example-s3-bucket1/object1")
                .destinationBucket("example-s3-bucket2")
                .destinationKey("object1copy")
                .expectedSourceBucketOwner("111122223333")
                .expectedBucketOwner("444455556666")
                .build();
        s3Client.copyObject(request);
    }
```

Example

Esempio: recuperare una policy del bucket

Nell'esempio seguente viene recuperata la policy di accesso per il bucket S3 *example-s3-bucket1*, utilizzando la condizione proprietario del bucket per assicurarsi che *example-s3-bucket1* sia di proprietà dell'account Account AWS 111122223333.

AWS CLI

```
aws s3api get-bucket-policy --bucket example-s3-bucket1 --expected-bucket-
owner 111122223333
```

AWS SDK for Java 2.x

```
public void getBucketPolicyExample() {
    S3Client s3Client = S3Client.create();
    GetBucketPolicyRequest request = GetBucketPolicyRequest.builder()
            .bucket("example-s3-bucket1")
            .expectedBucketOwner("111122223333")
            .build();
    try {
        GetBucketPolicyResponse response = s3Client.getBucketPolicy(request);
    }
    catch (S3Exception e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
}
```

Restrizioni e limitazioni

La condizione proprietario del bucket Amazon S3 presenta le seguenti restrizioni e limitazioni:

- Il valore del parametro della condizione del proprietario del bucket deve essere un Account AWS ID (valore numerico a 12 cifre). I principali del servizio non sono supportati.
- La condizione di proprietario del bucket non è disponibile per CreateBucketnessuna delle operazioni incluse in S3 Control. ListBucketsAWS Amazon S3 ignora tutti i parametri delle condizioni proprietario del bucket inclusi nelle richieste a queste operazioni.
- La condizione proprietario del bucket verifica solo che l'account specificato nel parametro di verifica possieda il bucket. La condizione proprietario del bucket non controlla la configurazione del bucket. Inoltre, non garantisce che la configurazione del bucket soddisfi condizioni specifiche o corrisponda a qualsiasi stato passato.

Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket

S3 Proprietà dell'oggetto è un'impostazione a livello di bucket Amazon S3 che è possibile utilizzare per controllare la proprietà degli oggetti caricati nel bucket e per disabilitare o abilitare le liste di controllo degli accessi (ACL). Per impostazione predefinita, Proprietà dell'oggetto è impostata su Proprietario del bucket applicato e tutte le ACL sono disabilitate. Quando le ACL sono disabilitate, il proprietario del bucket dispone di tutti gli oggetti nel bucket e gestisce l'accesso ai dati in maniera esclusiva utilizzando policy di gestione dell'accesso.

La maggior parte dei casi d'uso moderni in Amazon S3 non richiede più l'uso di ACL ed è consigliabile mantenere le ACL disabilitate tranne in circostanze straordinarie in cui è necessario controllare l'accesso individualmente per ciascun oggetto. Con le ACL disabilitate, puoi utilizzare le policy per controllare più facilmente l'accesso a tutti gli oggetti nel bucket, a prescindere da chi ha caricato gli oggetti nel bucket.

Object Ownership ha tre impostazioni che è possibile utilizzare per controllare la proprietà degli oggetti caricati nel bucket e disabilitare o abilitare le ACL:

ACL disabilitate

- Proprietario del bucket applicato (impostazione predefinita): le ACL sono disabilitate e il proprietario del bucket possiede automaticamente e ha il controllo completo di ogni oggetto nel bucket. Le

ACL non influiscono più sulle autorizzazioni per i dati nel bucket S3. Il bucket utilizza le policy per definire il controllo degli accessi.

ACL abilitate

- Proprietario del bucket scelto – Il proprietario del bucket possiede e ha il pieno controllo sui nuovi oggetti che altri account scrivono nel bucket con l'ACL predefinita `bucket-owner-full-control`.
- Object writer – L' Account AWS che carica un oggetto lo possiede, ne ha il pieno controllo e può concedere ad altri utenti l'accesso tramite ACL.

Per la maggior parte dei casi d'uso moderni in S3, è consigliabile mantenere le ACL disabilitate applicando l'impostazione Proprietario del bucket applicato e utilizzando la policy del bucket per condividere i dati con utenti esterni all'account in base alle esigenze. Questo approccio semplifica la gestione delle autorizzazioni. È possibile disabilitare le ACL su bucket appena creati e già esistenti. Per i bucket appena creati, le ACL sono disabilitate per impostazione predefinita. Nel caso di un bucket esistente che contiene già oggetti, dopo aver disabilitato le ACL, le ACL oggetto e bucket non faranno più parte di una valutazione dell'accesso, che verrà concesso o negato sulla base delle policy. Per i bucket esistenti, è possibile riattivare le ACL in qualsiasi momento dopo averle disabilitate e le ACL bucket e oggetto preesistenti verranno ripristinate.

Prima di disabilitare le ACL, ti consigliamo di rivedere la policy del bucket per assicurarti che copra tutti i modi in cui intendi concedere l'accesso al tuo bucket al di fuori del tuo account. Dopo aver disabilitato le ACL, il bucket accetterà solo richieste PUT che non specificano una ACL o richieste PUT con le ACL di controllo completo del proprietario del bucket, ad esempio l'ACL predefinita `bucket-owner-full-control` o forme equivalenti di questa ACL espresse in XML. Le applicazioni esistenti che supportano gli ACL di controllo completo del proprietario del bucket non hanno alcun impatto. PUTle richieste che contengono altri ACL (ad esempio, concessioni personalizzate a determinati Account AWS) hanno esito negativo e restituiscono un `400` errore con il codice di errore. `AccessControlListNotSupported`

Al contrario, un bucket con l'impostazione Bucket owner preferred continua ad accettare e rispettare le ACL di bucket e oggetti. Con questa impostazione, nuovi oggetti scritti con l'ACL predefinita `bucket-owner-full-control` saranno automaticamente di proprietà del proprietario del bucket anziché dell'object writer. Tutti gli altri comportamenti ACL rimangono in vigore. Per richiedere a tutte le operazioni PUT di Amazon S3 di includere l'ACL predefinita `bucket-owner-full-control`, puoi aggiungere una policy di bucket che consenta solo il caricamento di oggetti utilizzando questa ACL.

Per vedere quali impostazioni di Object Ownership vengono applicate ai tuoi bucket, puoi utilizzare i parametri di Amazon S3 Storage Lens. S3 Storage Lens è una funzionalità di analisi dell'archiviazione su cloud che puoi utilizzare per avere una panoramica completa a livello di organizzazione sull'utilizzo e sulle attività relative all'archiviazione di oggetti. Per ulteriori informazioni, consulta la sezione relativa all'utilizzo di S3 Storage Lens per trovare le impostazioni di Object Ownership.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Impostazioni di Object Ownership

Questa tabella mostra l'impatto di ogni impostazione Object Ownership su ACL, oggetti, proprietà di oggetti e caricamenti di oggetti.

Impostazione	Si applica a	Effetto sulla proprietà degli oggetti	Effetto sulle ACL	Caricamenti accettati
Proprietario del bucket applicato (impostazione predefinita)	Tutti gli oggetti esistenti e nuovi	Il proprietario del bucket possiede ogni oggetto.	Le ACL sono disabilitate e non influiscono più sulle autorizzazioni di accesso al bucket. Le richieste di impostazione o aggiornamento delle ACL sono respinte. Tuttavia, sono supportate le richieste di lettura delle ACL.	Caricamenti con ACL a controllo completo del proprietario del bucket o caricamenti che non specificano un'ACL

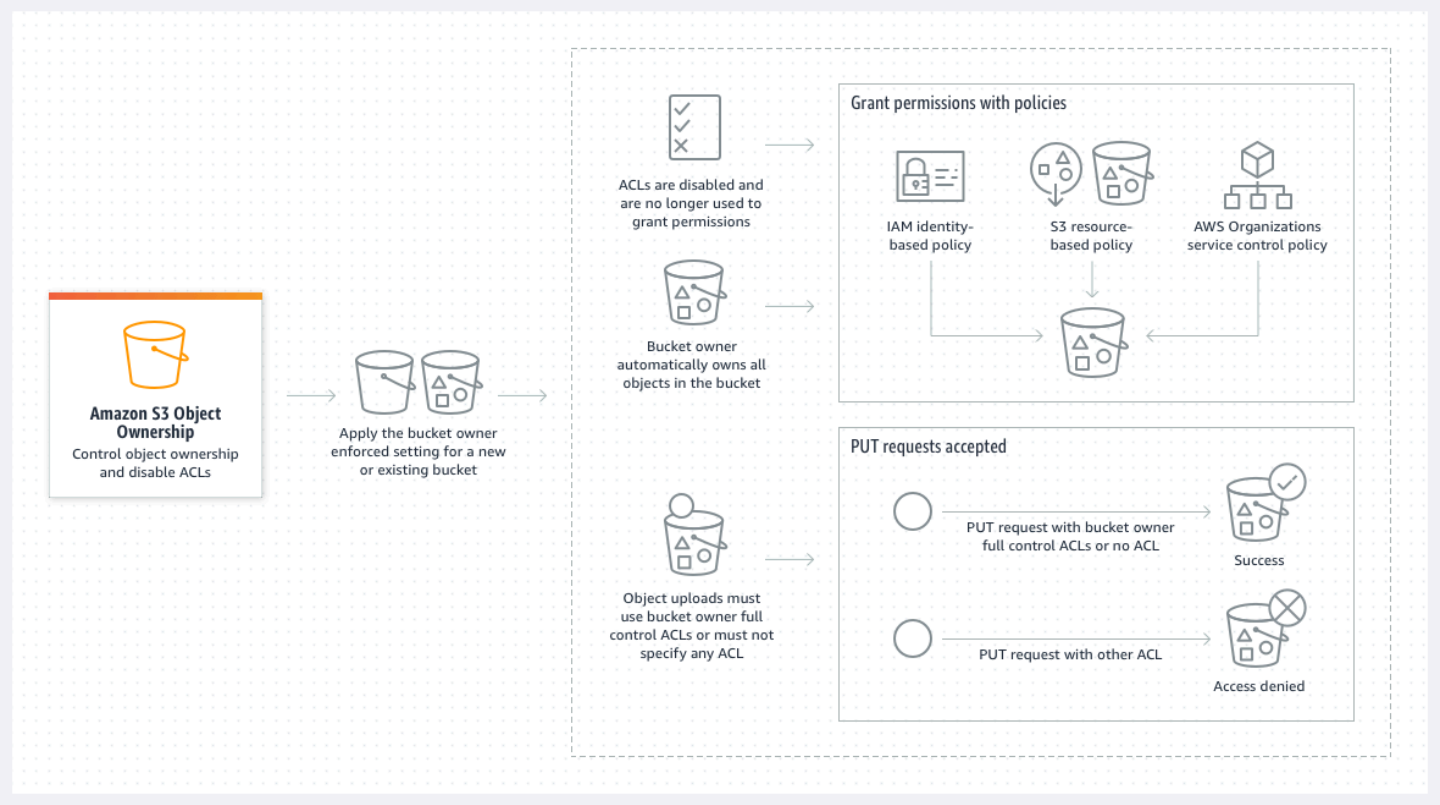
Impostazione	Si applica a	Effetto sulla proprietà degli oggetti	Effetto sulle ACL	Caricamenti accettati
			Il proprietario del bucket ha piena proprietà e controllo. L'object Writer non ha più piena proprietà e controllo.	
Proprietario del bucket preferito	Nuovi oggetti	Se un caricamento di oggetti include l'ACL preferita `bucket-owner-full-control`, il proprietario del bucket possiede l'oggetto. Gli oggetti caricati con altre ACL sono di proprietà dell'account di scrittura.	Le ACL possono essere aggiornate e possono concedere autorizzazioni. Se un caricamento di oggetti include l'ACL preferita `bucket-owner-full-control`, il proprietario del bucket ha completo controllo degli accessi mentre l'object writer non lo ha più.	Tutti i caricamenti

Impostazione	Si applica a	Effetto sulla proprietà degli oggetti	Effetto sulle ACL	Caricamenti accettati
Autore dell'oggetto	Nuovi oggetti	L'object writer è proprietario dell'oggetto.	Le ACL possono essere aggiornate e possono concedere autorizzazioni. L'Object writer ha completo controllo degli accessi.	Tutti i caricamenti

Modifiche introdotte disabilitando le ACL

Quando si applica l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto, le ACL vengono disabilitate e l'utente dispone automaticamente e acquisisce il controllo completo di tutti gli oggetti nel bucket senza eseguire alcuna azione aggiuntiva. Proprietario del bucket applicato è l'impostazione predefinita per tutti i nuovi bucket creati. Dopo aver applicato l'impostazione Proprietario del bucket applicato, verranno visualizzate tre modifiche:

- Tutte le ACL dei bucket e le ACL degli oggetti sono disabilitate, il che ti dà pieno accesso in quanto proprietario del bucket. Quando esegui una richiesta ACL di lettura sul bucket o sull'oggetto, vedrai che l'accesso completo è dato solo al proprietario del bucket.
- In quanto proprietario del bucket possiedi automaticamente e hai il pieno controllo su ogni oggetto nel bucket.
- Le ACL non influiscono più sulle autorizzazioni di accesso al bucket. Di conseguenza, il controllo degli accessi per i tuoi dati è basato su policy, come policy IAM, policy bucket S3, policy endpoint VPC e policy SCP di Organizations.



Se si utilizza il controllo delle versioni di S3, il proprietario del bucket possiede e ha il pieno controllo su tutte le versioni degli oggetti nel bucket. L'applicazione dell'impostazione Proprietario del bucket applicato non aggiunge una nuova versione di un oggetto.

I nuovi oggetti possono essere caricati nel bucket solo se utilizzano ACL a controllo completo del proprietario del bucket o non specificano un'ACL. I caricamenti di oggetti non riescono se specificano altre ACL. Per ulteriori informazioni, consulta Risoluzione dei problemi.

Dal momento che la seguente operazione esemplificativa `PutObject` che utilizza la AWS Command Line Interface (AWS CLI) include l'ACL predefinita `bucket-owner-full-control`, l'oggetto può essere caricato in un bucket con ACL disabilitate.

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key key-name --body path-to-file --
acl bucket-owner-full-control
```

Dal momento che la seguente operazione `PutObject` non specifica un'ACL, avrà esito positivo anche per un bucket con ACL disabilitate.

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key key-name --body path-to-file
```

> **Note**
>
> Se altri Account AWS hanno bisogno di accedere agli oggetti dopo il caricamento, devi concedere autorizzazioni aggiuntive a tali account tramite le policy dei bucket. Per ulteriori informazioni, consulta Procedure dettagliate che utilizzano policy per gestire l'accesso alle risorse Amazon S3.

Riabilitazione delle ACL

È possibile riabilitare le ACL passando dall'impostazione Proprietario del bucket applicato a un'altra impostazione di Proprietà dell'oggetto in qualsiasi momento. Se ACL oggetto sono state utilizzate per la gestione delle autorizzazioni prima di applicare l'impostazione Proprietario del bucket applicato e non è stata eseguita la migrazione delle autorizzazioni ACL dell'oggetto alla policy del bucket, dopo che le ACL vengono riabilitate, queste autorizzazioni verranno ripristinate. Inoltre, gli oggetti scritti nel bucket mentre era applicata l'impostazione Proprietario del bucket applicato, appartengono ancora al proprietario del bucket.

Ad esempio, se passi dall'impostazione Proprietario del bucket applicato all'impostazione Autore dell'oggetto, in qualità di proprietario del bucket, non disporrai più della proprietà e del controllo completo sugli oggetti che appartenevano in precedenza ad altri Account AWS. Al contrario, gli account di caricamento possiederanno nuovamente questi oggetti. Gli oggetti di proprietà di altri account utilizzano le ACL per le autorizzazioni, quindi non è possibile utilizzare le policy per concedere autorizzazioni a questi oggetti. Tuttavia, in qualità di proprietario del bucket, sei ancora il proprietario di tutti gli oggetti che sono stati scritti nel bucket mentre era applicata l'impostazione Proprietario del bucket applicato. Questi oggetti non sono di proprietà dell'object writer, anche se le ACL vengono riabilitate.

Per istruzioni su come abilitare e gestire gli ACL utilizzando AWS Management Console, AWS Command Line Interface (CLI), l'API REST AWS o gli SDK, consulta. Configurazione delle ACL

Prerequisiti per la disabilitazione delle ACL

Prima di disabilitare le ACL per un bucket esistente, è necessario soddisfare i seguenti prerequisiti.

Esamina le ACL di bucket e oggetti e migra le autorizzazioni ACL

Quando si disattivano le ACL, le autorizzazioni concesse dalle ACL di bucket e oggetti non influiranno più sull'accesso. Prima di disabilitare le ACL, rivedere le ACL bucket e oggetto.

Se le ACL del bucket concedono autorizzazioni di lettura o scrittura ad altri utenti esterni all'account, è necessario migrare queste autorizzazioni alla policy del bucket prima di poter applicare l'impostazione Proprietario del bucket applicato. Se non esegui la migrazione delle ACL del bucket che concedono l'accesso in lettura o scrittura al di fuori del tuo account, la richiesta di applicare l'impostazione Proprietario del bucket applicato non va a buon fine e restituisce il codice di errore InvalidBucketAclWithObjectOwnership.

Ad esempio, se si desidera disabilitare le ACL per un bucket che riceve i log di accesso al server, è necessario migrare le autorizzazioni ACL del bucket per il gruppo di recapito dei log S3 al principale del servizio di registrazione in una policy di bucket. Per ulteriori informazioni, consulta Concedere l'accesso al gruppo di consegna di log S3 per la registrazione di log degli accessi al server.

Se si desidera che l'object writer mantenga il pieno controllo dell'oggetto caricato, l'object writer è la migliore impostazione di Object Ownership per il caso d'uso. Se si desidera controllare l'accesso a livello di singolo oggetto, il proprietario del bucket preferito è la decisione migliore. Questi casi d'uso non sono comuni.

Per esaminare le ACL e migrare le autorizzazioni ACL alle policy del bucket, consultare Prerequisiti per la disabilitazione delle ACL.

Identifica tutte le richieste che richiedono una ACL per l'autorizzazione

Per identificare le richieste Amazon S3 che richiedono le ACL per l'autorizzazione, puoi utilizzare il valore `aclRequired` nei log degli accessi del server Amazon S3 oppure AWS CloudTrail. Se la richiesta richiede un ACL per l'autorizzazione o se sono presenti richieste PUT che specificano un ACL, la stringa è `Yes`. Se non era richiesto alcun ACL o se stai impostando un ACL predefinito o se le richieste sono consentite dalla tua policy del bucket, la stringa di `aclRequired` valore è "" nei log di accesso al server di - Amazon S3 `bucket-owner-full-control` ed è assente in. CloudTrail Per ulteriori informazioni sui valori `aclRequired` attesi, consulta Valori `aclRequired` per le richieste di Amazon S3.

Se hai richieste `PutBucketAcl` o `PutObjectAcl` con intestazioni che concedono autorizzazioni basate sugli ACL, ad eccezione degli ACL `bucket-owner-full-control` predefiniti, devi rimuovere tali intestazioni prima di poter disabilitare gli ACL. In caso contrario, le tue richieste non avranno esito positivo.

Per tutte le altre richieste che richiedevano un ACL per l'autorizzazione, migra tali autorizzazioni ACL alle policy dei bucket. Quindi, rimuovi tutti gli ACL del bucket prima di abilitare l'impostazione forzata del proprietario del bucket.

> **Note**
>
> Non rimuovere gli ACL degli oggetti. In caso contrario, le applicazioni che si basano sugli ACL degli oggetti per le autorizzazioni perderanno l'accesso.

Se vedi che nessuna richiesta richiedeva un ACL per l'autorizzazione, puoi procedere alla disattivazione degli ACL. Per ulteriori informazioni sull'identificazione delle richieste, vedere Utilizzo dei log degli accessi al server Amazon S3 per identificare le richieste eIdentificazione delle richieste Amazon S3 tramite CloudTrail.

Esamina e aggiorna le policy del bucket che utilizzano chiavi di condizione relative all'ACL

Dopo aver applicato l'impostazione Proprietario del bucket applicato per disabilitare le ACL, i nuovi oggetti possono essere caricati nel bucket solo se la richiesta utilizza ACL di controllo completo del proprietario del bucket o se non specifica un'ACL. Prima di disabilitare le ACL, consulta la policy del bucket per le chiavi di condizione relative all'ACL.

Se la policy del bucket utilizza una chiave di condizione relativa all'ACL per richiedere l'ACL predefinita `bucket-owner-full-control` (ad esempio `s3:x-amz-acl`), non è necessario aggiornare la policy del bucket. La seguente policy di bucket utilizza il codice `s3:x-amz-acl` per richiedere l'ACL predefinita `bucket-owner-full-control` per le richieste `PutObject` di S3. Questa policy richiede ancora all'object writer di specificare l'ACL predefinita `bucket-owner-full-control`. Tuttavia, i bucket con ACL disabilitate accettano ancora questa ACL, quindi le richieste continuano ad avere successo senza bisogno di modifiche sul lato client.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Only allow writes to my bucket with bucket owner full control",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/ExampleUser"
                ]
            },
            "Action": [
                "s3:PutObject"
            ],
```

```
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
```

Tuttavia, se la policy di bucket utilizza una chiave di condizione relativa all'ACL che richiede un'ACL diversa, è necessario rimuovere questa chiave di condizione. Questo esempio di policy di bucket richiede l'ACL `public-read` per le richieste `PutObject` di S3 e quindi deve essere aggiornata prima di disabilitare le ACL.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Only allow writes to my bucket with public read access",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/ExampleUser"                ]
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "public-read"
                }
            }
        }
    ]
}
```

Autorizzazioni di Object Ownership

Per applicare, aggiornare o eliminare un'impostazione di Object Ownership per un bucket, è necessaria l'autorizzazione `s3:PutBucketOwnershipControls`. Per

restituire l'impostazione Object Ownership per un bucket, è necessaria l'autorizzazione `s3:GetBucketOwnershipControls`. Per ulteriori informazioni, consulta Impostazione di Object Ownership quando si crea un bucket e Visualizzare l'impostazione di Object Ownership per un bucket S3.

Disabilitare le ACL per tutti i nuovi bucket

Per impostazione predefinita, tutti i nuovi bucket vengono creati con l'impostazione Proprietario del bucket applicato applicata e le ACL sono disabilitate. È consigliabile mantenere le ACL disabilitate. Come regola generale, è consigliabile utilizzare policy basate sulle risorse S3 (policy di bucket e policy dei punti di accesso) o policy IAM per il controllo degli accessi anziché ACL. Le policy sono un'opzione di controllo degli accessi semplificata e più flessibile. Con le policy dei bucket e le policy dei punti di accesso, puoi definire le regole applicabili globalmente a tutte le richieste alle risorse Amazon S3.

Replication e Object Ownership

Quando utilizzi la replica S3 e i bucket di origine e di destinazione sono di proprietà di diversi Account AWS, puoi disabilitare gli ACL (con l'impostazione applicata dal proprietario del bucket per Object Ownership) per modificare la proprietà della replica con Account AWS quella che possiede il bucket di destinazione. Questa impostazione imita il comportamento di sovrascrittura del proprietario esistente senza la necessità di un'autorizzazione `s3:ObjectOwnerOverrideToBucketOwner`. Tutti gli oggetti replicati nel bucket di destinazione con l'impostazione Proprietario del bucket applicato sono di proprietà del proprietario del bucket di destinazione. Per ulteriori informazioni sull'opzione di sovrascrittura del proprietario per le configurazioni di replica, consulta Modifica del proprietario della replica.

Impostazione di Object Ownership

Puoi applicare un'impostazione di proprietà degli oggetti utilizzando la console Amazon S3, gli AWS SDK AWS CLI, l'API REST di Amazon S3 oppure. AWS CloudFormation Le seguenti API REST e i seguenti AWS CLI comandi supportano Object Ownership:

REST API	AWS CLI	Descrizione
PutBucketOwnershipControls	put-bucket-ownership-controls	Crea o modifica l'impostazione Object Ownership per un bucket S3 esistente.

REST API	AWS CLI	Descrizione
CreateBucket	create-bucket	Crea un bucket tramite l'intestazione `x-amz-object-ownership` della richiesta per specificare l'impostazione Object Ownership.
GetBucketOwnershipControls	get-bucket-ownership-controls	Recupera l'impostazione Object Ownership per un bucket Amazon S3.
DeleteBucketOwnershipControls	delete-bucket-ownership-controls	Elimina l'impostazione Object Ownership per un bucket Amazon S3.

Per ulteriori informazioni sull'applicazione e l'utilizzo delle impostazioni di Object Ownership, consultare gli argomenti riportati di seguito.

Argomenti

- Prerequisiti per la disabilitazione delle ACL
- Impostazione di Object Ownership quando si crea un bucket
- Impostazione di Object Ownership su un bucket esistente
- Visualizzare l'impostazione di Object Ownership per un bucket S3
- Disabilitare le ACL per tutti i nuovi bucket e applicare Object Ownership
- Risoluzione dei problemi

Prerequisiti per la disabilitazione delle ACL

Se l'ACL del bucket concede l'accesso all'esterno del bucket Account AWS, prima di disabilitare gli ACL, è necessario migrare le autorizzazioni ACL del bucket alla policy del bucket e ripristinare l'ACL del bucket sull'ACL privato predefinito. Se non esegui la migrazione di queste ACL bucket, la richiesta per applicare l'impostazione Proprietario del bucket applicato per disabilitare le ACL non va a buon fine e restituisce il codice di errore InvalidBucketAclWithObjectOwnership. Ti consigliamo inoltre di

rivedere le autorizzazioni ACL dell'oggetto e di migrarle alla policy di bucket. Per ulteriori informazioni su altri prerequisiti consigliati, consulta Prerequisiti per la disabilitazione delle ACL.

Ciascuna delle ACL di bucket e di oggetti esistenti ha un equivalente in una policy IAM. I seguenti esempi di policy di bucket mostrano come le autorizzazioni `READ` e `WRITE` per le ACL di bucket e di oggetti sono associate alle autorizzazioni IAM. Per ulteriori informazioni su come ogni ACL si traduce in autorizzazioni IAM, consulta Mappatura delle autorizzazioni ACL e delle autorizzazioni della policy di accesso.

Per esaminare e migrare le autorizzazioni ACL alle policy di bucket, consultare i seguenti argomenti.

Argomenti

- Esempi di policy di bucket
- Utilizzo della console S3 per esaminare e migrare le autorizzazioni ACL
- Utilizzo di AWS CLI per rivedere e migrare le autorizzazioni ACL
- Procedure guidate di esempio

Esempi di policy di bucket

Queste policy di bucket esemplificative mostrano come migrare le autorizzazioni ACL `READ` e `WRITE` di bucket e di oggetti per un Account AWS di terze parti ad una policy di bucket. Le ACL `READ_ACP` e `WRITE_ACP` sono meno rilevanti per le policy perché concedono autorizzazioni relative all'ACL (`s3:GetBucketAcl`,`s3:GetObjectAcl`,`s3:PutBucketAcl`, `es3:PutObjectAcl`).

Example — **`READ`** ACL per un bucket

Se il tuo bucket dispone di un `READ` ACL che concede l' Account AWS *111122223333*autorizzazione a elencare i contenuti del bucket, puoi scrivere una policy del bucket che conceda le autorizzazioni per il bucket. `s3:ListBucket` `s3:ListBucketVersions` `s3:ListBucketMultipartUploads`

```
{
    "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** per ogni oggetto in un bucket

Se ogni oggetto nel tuo bucket ha un READ ACL a cui concede l'accesso Account AWS *111122223333*, puoi scrivere una policy del bucket che `s3:GetObject` conceda e autorizzi a questo account per ogni oggetto nel tuo bucket. `s3:GetObjectVersion`

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

Questo elemento di risorsa esemplificativo consente l'accesso a un oggetto specifico.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/OBJECT-KEY"
```

Example — **`WRITE`** ACL che concede le autorizzazioni per scrivere oggetti su un bucket

Se il tuo bucket ha un `WRITE` ACL che concede l' Account AWS *111122223333*autorizzazione a scrivere oggetti nel tuo bucket, puoi scrivere una policy sul bucket che conceda l'autorizzazione per il tuo bucket. `s3:PutObject`

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

Utilizzo della console S3 per esaminare e migrare le autorizzazioni ACL

Per esaminare le autorizzazioni ACL di un bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket, seleziona il nome del bucket.
3. Scegli la scheda Autorizzazioni.
4. Alla voce Lista di controllo accessi (ACL), controlla le autorizzazioni ACL del bucket.

Per esaminare le autorizzazioni ACL di un oggetto

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket), scegli il nome del bucket contenente l'oggetto.

3. Nell'elenco Oggetti scegli il nome dell'oggetto.
4. Scegli la scheda Autorizzazioni.
5. Alla voce Lista di controllo accessi (ACL), controlla le autorizzazioni ACL dell'oggetto.

Per migrare le autorizzazioni ACL e aggiornare l'ACL del bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket, seleziona il nome del bucket.
3. Nella sezione Autorizzazioni, alla voce Policy del bucket, scegliere Modifica.
4. Nella casella Policy, aggiungi o aggiorna la policy del bucket.

   Per le policy di bucket di esempio, consulta Esempi di policy di bucket e Procedure guidate di esempio.
5. Seleziona Salvataggio delle modifiche.
6. Aggiorna l'ACL del bucket per rimuovere le autorizzazioni ACL ad altri gruppi o Account AWS.
7. Applica l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto.

Utilizzo di AWS CLI per rivedere e migrare le autorizzazioni ACL

1. Per restituire l'ACL del bucket per il tuo bucket, usa il comando: get-bucket-acl AWS CLI

   ```
   aws s3api get-bucket-acl --bucket DOC-EXAMPLE-BUCKET
   ```

   Ad esempio, questa ACL di bucket concede l'accesso `WRITE` e `READ` a un account di terze parti. In questa ACL, l'account di terze parti è identificato dall'ID utente canonico. Per applicare l'impostazione Proprietario del bucket applicato e disabilitare le ACL, è necessario migrare queste autorizzazioni per l'account di terze parti a una policy del 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"
        }
    ]
}
```

Per altre ACL di esempio, consultare Procedure guidate di esempio.

2. Migrazione delle autorizzazioni ACL del bucket a una policy di bucket:

   Questo esempio di policy di bucket concede autorizzazioni `s3:PutObject` e `s3:ListBucket` per un account di terze parti. Nella policy bucket, l'account di terze parti è identificato dall'ID (). Account 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/*"
        }
    ]
}
```

Per ulteriori policy di bucket esemplificative, consulta Esempi di policy di bucket e Procedure guidate di esempio.

3. Per restituire l'ACL per un oggetto specifico, utilizzate il get-object-acl AWS CLI comando.

```
aws s3api get-object-acl --bucket DOC-EXAMPLE-BUCKET --key EXAMPLE-OBJECT-KEY
```

4. Se necessario, migrare le autorizzazioni ACL degli oggetti alla policy del bucket.

   Questo elemento di risorsa esemplificativo concede l'accesso a un oggetto specifico in una policy di bucket.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/EXAMPLE-OBJECT-KEY"
```

5. Ripristina l'ACL per il bucket sull'ACL predefinito.

```
aws s3api put-bucket-acl --bucket DOC-EXAMPLE-BUCKET --acl private
```

6. Applica l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto.

Procedure guidate di esempio

Negli esempi seguenti viene illustrato come migrare le autorizzazioni ACL alle policy di bucket per casi d'uso specifici.

Argomenti

- Concedere l'accesso al gruppo di consegna di log S3 per la registrazione di log degli accessi al server
- Concedere l'accesso pubblico in lettura agli oggetti nel bucket
- Concedi ad Amazon ElastiCache for Redis l'accesso al tuo bucket S3

Concedere l'accesso al gruppo di consegna di log S3 per la registrazione di log degli accessi al server

Se desideri applicare l'impostazione Proprietario del bucket applicato per disabilitare le ACL per un bucket di destinazione di registrazione di log degli accessi a un server (noto anche come bucket target), è necessario migrare le autorizzazioni ACL bucket per il gruppo di consegna di log S3 al principale del servizio di registrazione di log (`logging.s3.amazonaws.com`) in una policy del bucket. Per ulteriori informazioni sulle autorizzazioni della distribuzione dei registri, consultare Autorizzazioni per la distribuzione dei registri.

Questa ACL del bucket concede l'accesso `WRITE` e `READ_ACP` al gruppo di distribuzione di registri 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"
        }
    ]
}
```

Per migrare le autorizzazioni ACL del bucket per il gruppo di distribuzione di registri S3 al principale del servizio di registrazione in una policy di bucket

1. Aggiungi la seguente policy di bucket al bucket di destinazione, sostituendo i valori di esempio.

```
aws s3api put-bucket-policy --bucket DOC-EXAMPLE-BUCKET --policy file://policy.json

policy.json:      {
    {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ServerAccessLogsPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/EXAMPLE-LOGGING-PREFIX*",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::SOURCE-BUCKET-NAME"
                },
                "StringEquals": {
                    "aws:SourceAccount": "SOURCE-AWS-ACCOUNT-ID"
                }
            }
        }
    ]
```

```
}
```

2. Ripristina l'ACL per il bucket di destinazione all'ACL predefinita.

```
aws s3api put-bucket-acl --bucket DOC-EXAMPLE-BUCKET --acl private
```

3. Applica l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto al bucket di destinazione.

Concedere l'accesso pubblico in lettura agli oggetti nel bucket

Se le ACL di oggetto consentono l'accesso pubblico in lettura a tutti gli oggetti del bucket, è possibile migrare queste autorizzazioni ACL a una policy di bucket.

Questa ACL di oggetto concede l'accesso pubblico in lettura a un oggetto in 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"
        }
    ]
}
```

Per migrare le autorizzazioni ACL di lettura pubblica a una policy di bucket

1. Per concedere l'accesso in lettura pubblica a tutti gli oggetti nel bucket, aggiungere la seguente policy di bucket, sostituendo i valori di esempio.

```
aws s3api put-bucket-policy --bucket DOC-EXAMPLE-BUCKET --policy file://policy.json

policy.json:
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ]
        }
    ]
}
```

   Per concedere l'accesso pubblico a un oggetto specifico in una policy di bucket, utilizzare il seguente formato per l'elemento `Resource`.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/OBJECT-KEY"
```

   Per concedere l'accesso pubblico a tutti gli oggetti con un prefisso specifico, utilizzare il seguente formato per l'elemento `Resource`.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/PREFIX/*"
```

2. Applica l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto.

Concedi ad Amazon ElastiCache for Redis l'accesso al tuo bucket S3

Puoi esportare il tuo backup ElastiCache per Redis in un bucket S3, che ti consente di accedere al backup dall'esterno. ElastiCache Per esportare il backup in un bucket S3, devi concedere le ElastiCache autorizzazioni per copiare un'istantanea nel bucket. Se hai concesso le autorizzazioni a un ACL ElastiCache in un bucket, devi migrare queste autorizzazioni a una policy del bucket prima di applicare l'impostazione applicata dal proprietario del bucket per disabilitare gli ACL. Per ulteriori informazioni, consulta Concedi ElastiCache l'accesso al tuo bucket Amazon S3 nella Amazon ElastiCache User Guide.

L'esempio seguente mostra le autorizzazioni ACL del bucket a cui concedono le autorizzazioni. 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"
        }
    ]
}
```

Per migrare le autorizzazioni ACL del bucket per Redis a una policy bucket ElastiCache

1. Aggiungere la seguente policy di bucket al bucket di destinazione, sostituendo i valori di esempio.

```
aws s3api put-bucket-policy --bucket DOC-EXAMPLE-BUCKET --policy file://policy.json

policy.json:
"Id": "Policy15397346",
    "Statement": [
        {
            "Sid": "Stmt15399483",
            "Effect": "Allow",
            "Principal": {
                "Service": "Region.elasticache-snapshot.amazonaws.com"
            },
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:GetBucketAcl",
                "s3:ListMultipartUploadParts",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
```

```
            ]
        }
    ]
}
```

2. Resettare l'ACL per il bucket all'ACL di default:

```
aws s3api put-bucket-acl --bucket DOC-EXAMPLE-BUCKET --acl private
```

3. Applica l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto.

Impostazione di Object Ownership quando si crea un bucket

Quando crei un bucket, puoi configurare S3 Object Ownership. Per impostare Object Ownership per un bucket esistente, consultare Impostazione di Object Ownership su un bucket esistente.

S3 Object Ownership è un'impostazione a livello di bucket di Amazon S3 che puoi utilizzare per disabilitare le liste di controllo accessi (ACL) e assumere la proprietà di ogni oggetto nel tuo bucket, semplificando la gestione degli accessi per i dati archiviati in Amazon S3. Per impostazione predefinita, Proprietà dell'oggetto S3 è impostata su Proprietario del bucket applicato e le ACL sono disabilitate per nuovi bucket. Con le ACL disabilitate, il proprietario del bucket possiede ogni oggetto nel bucket e gestisce l'accesso ai dati in maniera esclusiva utilizzando policy di gestione dell'accesso. È consigliabile mantenere le ACL disabilitate, tranne in circostanze insolite in cui è necessario controllare individualmente l'accesso per ciascun oggetto.

Object Ownership ha tre impostazioni che è possibile utilizzare per controllare la proprietà degli oggetti caricati nel bucket e disabilitare o abilitare le ACL:

ACL disabilitate

- Proprietario del bucket applicato (impostazione predefinita): le ACL sono disabilitate e il proprietario del bucket possiede automaticamente e ha il controllo completo di ogni oggetto nel bucket. Le ACL non influiscono più sulle autorizzazioni per i dati nel bucket S3. Il bucket utilizza le policy per definire il controllo degli accessi.

ACL abilitate

- Proprietario del bucket scelto – Il proprietario del bucket possiede e ha il pieno controllo sui nuovi oggetti che altri account scrivono nel bucket con l'ACL predefinita `bucket-owner-full-control`.

- Object writer: chi carica un oggetto possiede l' Account AWS oggetto, ne ha il pieno controllo e può concedere ad altri utenti l'accesso ad esso tramite ACL.

Autorizzazioni: per applicare l'impostazione Bucket owner enforced (Applicata da proprietario bucket) oppure Bucket owner preferred (Preferita da proprietario bucket), devi disporre delle seguenti autorizzazioni: `s3:CreateBucket` e `s3:PutBucketOwnershipControls`. Non sono necessarie autorizzazioni aggiuntive quando si crea un bucket con l'impostazione Object writer applicata. Per ulteriori informazioni sulle autorizzazioni di Amazon S3, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.

> **Important**
>
> La maggior parte dei casi d'uso moderni in Amazon S3 non richiede più l'uso di ACL ed è consigliabile disabilitarle tranne in circostanze straordinarie in cui è necessario controllare l'accesso individualmente per ciascun oggetto. Con Object Ownership, è possibile disabilitare le ACL e fare affidamento sulle policy per il controllo degli accessi. Quando disabiliti gli ACL, puoi gestire facilmente un bucket con oggetti caricati da account diversi. AWS In qualità di proprietario del bucket, possiedi tutti gli oggetti nel bucket e puoi gestirne l'accesso utilizzando le policy.

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nella barra di navigazione nella parte superiore della pagina, scegli il nome del file attualmente visualizzato Regione AWS. Quindi, scegli la regione in cui desideri creare un bucket.

   > **Note**
   >
   > Scegli una regione nelle tue vicinanze per ridurre al minimo la latenza e i costi o essere conforme ai requisiti normativi. Gli oggetti archiviati in una regione non la lasciano mai a meno che non vengano trasferiti esplicitamente in un'altra regione. Per un elenco di Amazon S3 Regioni AWS, consulta gli Servizio AWS endpoint in. Riferimenti generali di Amazon Web Services

3. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).

4. Scegliere Create bucket (Crea bucket).

   Viene visualizzata la pagina Create bucket (Crea bucket).

5. In Configurazione generale, visualizza Regione AWS dove verrà creato il bucket.

6. In Tipo di bucket, scegli Scopo generale.

7. In Nome bucket, immettere il nome del bucket.

   Il nome del bucket deve:

   - Essere univoco all'interno di una partizione. Una partizione è un raggruppamento di regioni. AWS ha attualmente tre partizioni: `aws` (regioni standard), `aws-cn` (regioni Cina) e `aws-us-gov` (AWS GovCloud (US) Regions).
   - Deve contenere da 3 a 63 caratteri
   - Essere costituito solo da lettere minuscole, numeri, punti (.) e trattini (-). Per una migliore compatibilità, si consiglia di evitare l'utilizzo di punti (.) nei nomi dei bucket, ad eccezione dei bucket utilizzati solo per l'hosting di siti Web statici.
   - Iniziare e finire con una lettera o un numero.

   Una volta creato il bucket, non è possibile modificarne il nome. Per ulteriori informazioni sulla denominazione dei bucket, consulta Regole di denominazione dei bucket.

   > **Important**
   >
   > Evitare di includere informazioni riservate, ad esempio numeri di account, nel nome del bucket. Il nome bucket è visibile nell'URL che punta agli oggetti nel bucket.

8. AWS Management Console ti consente di copiare le impostazioni di un bucket esistente nel tuo nuovo bucket. Se non desideri copiare le impostazioni di un bucket esistente, vai al passaggio successivo.

   > **Note**
   >
   > Questa opzione:
   >
   > - non è disponibile in AWS CLI ed è disponibile solo nella console
   > - Non è disponibile per i bucket di directory

- Non copia la policy del bucket dal bucket esistente al nuovo bucket

Per copiare le impostazioni di un bucket esistente, in Copia le impostazioni dal bucket esistente, seleziona Scegli il bucket. Si apre la finestra Scegli il bucket. Trova il bucket con le impostazioni che desideri copiare e seleziona Scegli il bucket. La finestra Scegli il bucket si chiude e la finestra Crea bucket si riapre.

In Copia le impostazioni dal bucket esistente, ora vedrai il nome del bucket selezionato. Vedrai anche l'opzione Ripristina i valori predefiniti che puoi usare per rimuovere le impostazioni del bucket copiato. Controlla le impostazioni rimanenti del bucket, nella pagina Crea bucket. Vedrai che ora corrispondono alle impostazioni del bucket che hai selezionato. Puoi passare alla fase finale.

9. Alla voce Proprietà oggetto, per disabilitare o abilitare le ACL e controllare la proprietà degli oggetti caricati nel bucket, scegliere una delle seguenti impostazioni:

   ACL disabilitate

   - Proprietario del bucket applicato (impostazione predefinita): le ACL sono disabilitate e il proprietario del bucket possiede automaticamente e ha il controllo completo di ogni oggetto nel bucket. Le liste di controllo degli accessi (ACL) non influiscono più sulle autorizzazioni di accesso ai dati nel bucket S3. Il bucket utilizza le policy esclusivamente per definire il controllo degli accessi.

     Per impostazione predefinita, le ACL sono disabilitate. La maggior parte degli attuali casi d'uso in Amazon S3 non richiede più l'uso delle ACL. È consigliabile mantenere le ACL disabilitate, tranne in circostanze insolite in cui è necessario controllare individualmente l'accesso per ciascun oggetto. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

   ACL abilitate

   - Proprietario del bucket scelto – Il proprietario del bucket possiede e ha il pieno controllo sui nuovi oggetti che altri account scrivono nel bucket con l'ACL predefinita `bucket-owner-full-control`.

     Se applichi l'impostazione Proprietario del bucket preferito, per richiedere che tutti i caricamenti di Amazon S3 includano l'ACL predefinita `bucket-owner-full-control`, puoi

aggiungere una policy del bucket che consenta solo il caricamento di oggetti che utilizzano questa ACL.

- Scrittore di oggetti: chi carica un oggetto possiede l'oggetto, ne ha il pieno controllo e può concedere ad altri utenti l'accesso ad esso tramite ACL. Account AWS

> **Note**
>
> L'impostazione predefinita è Proprietario del bucket applicato. Per applicare l'impostazione predefinita e mantenere gli ACL disabilitati, è necessaria solo l'autorizzazione `s3:CreateBucket`. Per abilitare gli ACL, è necessario disporre dell'autorizzazione `s3:PutBucketOwnershipControls`.

10. In Impostazioni di blocco dell'accesso pubblico per questo bucket scegli le impostazioni di blocco dell'accesso pubblico che vuoi applicare al bucket.

    Per impostazione predefinita, tutte e quattro le impostazioni Blocco dell'accesso pubblico sono abilitate. È consigliabile mantenere tutte le impostazioni abilitate, a meno che non sia necessario disattivarne una o più di una per il caso d'uso specifico. Per ulteriori informazioni sul blocco dell'accesso pubblico, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

    > **Note**
    >
    > Per abilitare tutte le impostazioni Blocco dell'accesso pubblico, è richiesta solo l'autorizzazione `s3:CreateBucket`. Per disattivare le impostazioni Blocco dell'accesso pubblico, è necessario disporre dell'autorizzazione `s3:PutBucketPublicAccessBlock`.

11. (Facoltativo) In Bucket Versioning (Controllo delle versioni bucket), puoi scegliere se conservare varianti degli oggetti nel bucket. Per ulteriori informazioni sulla funzione Controllo delle versioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

    Per disabilitare o abilitare il controllo delle versioni nel bucket, scegli Disable (Disabilita) o Enable (Abilita).

12. (Facoltativo) In Tags (Tag), puoi scegliere di aggiungere tag al bucket. I tag sono coppie chiave-valore utilizzate per classificare lo spazio di archiviazione.

Per aggiungere un tag al bucket, inserisci un valore in Key (Chiave) e facoltativamente un valore in Value (Valore), quindi scegli Add Tag (Aggiungi tag).

13. In Default encryption (Crittografia di default), scegliere Edit (Modifica).
14. Per configurare la crittografia predefinita, in Tipo di crittografia scegli una delle seguenti opzioni:

    - Chiavi gestite Amazon S3 (SSE-S3)
    - AWS Key Management Service chiave (SSE-KMS)

    > **Important**
    >
    > Se usi l'opzione SSE-KMS per la configurazione della crittografia predefinita, sei soggetto alla quota delle richieste al secondo di AWS KMS. Per ulteriori informazioni sulle AWS KMS quote e su come richiedere un aumento delle quote, consulta Quotas nella Developer Guide.AWS Key Management Service

    I bucket e i nuovi oggetti sono crittografati con la crittografia lato server con una chiave gestita da Amazon S3 come livello base di configurazione della crittografia. Per ulteriori informazioni sulla crittografia predefinita, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3.

    Per ulteriori informazioni sull'utilizzo della crittografia lato server di Amazon S3 per crittografare i dati, consulta Uso della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).
15. Se scegli Chiave AWS Key Management Service (SSE-KMS), procedi come segue:

    a. In Chiave AWS KMS specifica la tua chiave KMS in uno dei seguenti modi:

       - Per scegliere da un elenco di chiavi KMS disponibili, scegli tra le tue AWS KMS keys chiavi KMS e scegli la tua chiave KMS dall'elenco delle chiavi disponibili.

         In questo elenco vengono visualizzate sia la chiave Chiave gestita da AWS (`aws/s3`) che quella gestita dai clienti. Per ulteriori informazioni sulle chiavi gestite dal cliente, consulta Chiavi gestite dal cliente e chiavi AWS nella Guida per gli sviluppatori di AWS Key Management Service .
       - Per specificare l'ARN della chiave KMS, scegli Inserisci l'ARN della AWS KMS key e quindi specifica l'ARN della chiave KMS nel campo visualizzato.

- Per creare una nuova chiave gestita dal cliente nella AWS KMS console, scegli Crea una chiave KMS.

  Per ulteriori informazioni sulla creazione di una AWS KMS key, consulta Creating keys nella AWS Key Management Service Developer Guide.

> **Important**
>
> Puoi utilizzare solo le chiavi KMS disponibili nello Regione AWS stesso bucket. La console Amazon S3 elenca solo le prime 100 chiavi KMS nella stessa regione del bucket. Per utilizzare una chiave KMS non elencata, devi inserire l'ARN della chiave KMS. Se desideri utilizzare una chiave KMS di proprietà di un account diverso, è necessario innanzitutto disporre dell'autorizzazione necessaria per l'uso della chiave e quindi inserire l'ARN della chiave KMS. Per ulteriori informazioni sulle autorizzazioni tra account per le chiavi KMS, consulta Creazione di chiavi KMS utilizzabili da altri account nella Guida per gli sviluppatori di AWS Key Management Service . Per ulteriori informazioni su SSE-KMS, consulta Specifica della crittografia lato server con AWS KMS (SSE-KMS).
> Quando utilizzi una AWS KMS key crittografia lato server in Amazon S3, devi scegliere una chiave KMS di crittografia simmetrica. Amazon S3 supporta solo chiavi KMS di crittografia simmetriche e non chiavi KMS asimmetriche. Per ulteriori informazioni, consulta Identificazione delle chiavi KMS simmetriche e asimmetriche nella Guida per gli sviluppatori di AWS Key Management Service .

  Per ulteriori informazioni sulla creazione di una AWS KMS key, consulta Creating keys nella Developer Guide.AWS Key Management Service Per ulteriori informazioni sull'utilizzo AWS KMS con Amazon S3, consulta. Utilizzo della crittografia lato server con chiavi (SSE-KMS) AWS KMS

b. Quando configuri il bucket per utilizzare la crittografia predefinita con SSE-KMS puoi anche abilitare le chiavi bucket S3. S3 Bucket Keys riduce il costo della crittografia diminuendo il traffico di richieste da Amazon S3 a. AWS KMS Per ulteriori informazioni, consulta Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3.

   Per utilizzare le chiavi bucket S3, in Chiave bucket seleziona Abilita.

16. (Facoltativo) Se si desidera abilitare il blocco oggetti S3, effettua le seguenti operazioni:

a. Scegli Impostazioni avanzate.

> **Important**
>
> L'abilitazione del blocco oggetti consente anche la funzione Controllo delle versioni del bucket. Dopo averlo abilitato, per il blocco di oggetti è necessario configurare le impostazioni predefinite di conservazione e di blocco di carattere legale per proteggere i nuovi oggetti dall'eliminazione o dalla sovrascrittura.

b. Se desideri abilitare il blocco degli oggetti, scegli Enable (Abilita), leggi l'avviso visualizzato e confermalo.

Per ulteriori informazioni, consulta Utilizzo del blocco oggetti S3.

> **Note**
>
> Per creare un bucket abilitato per il blocco degli oggetti, devi disporre delle seguenti autorizzazioni: `s3:CreateBucket`, `s3:PutBucketVersioning` e `s3:PutBucketObjectLockConfiguration`.

17. Seleziona Crea bucket.

Usando il AWS CLI

Per impostare la proprietà dell'oggetto quando create un nuovo bucket, utilizzate il `create-bucket` AWS CLI comando con il `--object-ownership` parametro.

In questo esempio viene applicata l'impostazione Proprietario del bucket applicato per un nuovo bucket utilizzando la AWS CLI:

```
aws s3api create-bucket --bucket  DOC-EXAMPLE-BUCKET --region us-east-1 --object-
ownership BucketOwnerEnforced
```

> **Important**
>
> Se non imposti la proprietà dell'oggetto quando crei un bucket utilizzando il AWS CLI, l'impostazione predefinita sarà `ObjectWriter` (ACL abilitati).

Utilizzo dell' AWS SDK for Java

In questo esempio viene definita l'impostazione Proprietario del bucket applicato per un nuovo bucket utilizzando 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);
```

Usando AWS CloudFormation

Per utilizzare la `AWS::S3::Bucket` AWS CloudFormation risorsa per impostare la proprietà degli oggetti quando crei un nuovo bucket, consulta OwnershipControlsla AWS::S3::Bucket Guida per l'AWS CloudFormation utente.

Utilizzo di REST API

Per applicare l'impostazione Proprietario del bucket applicato per S3 Proprietà dell'oggetto, utilizza l'operazione API `CreateBucket` con l'intestazione della richiesta `x-amz-object-ownership` impostata su `BucketOwnerEnforced`. Per ulteriori informazioni, consulta CreateBucket nella Guida di riferimento per l'API di Amazon Simple Storage Service.

Fasi successive: dopo aver eseguito le impostazioni Proprietario del bucket applicato o Proprietario del bucket preferito per Proprietà dell'oggetto, è possibile compiere i seguenti passaggi:

- Proprietario del bucket applicato – Richiedi che tutti i nuovi bucket vengano creati con ACL disabilitate utilizzando le policy IAM o di Organizations.
- Proprietario del bucket preferito – Aggiungi una policy di bucket S3 per richiedere l'ACL predefinita `bucket-owner-full-control` per tutti gli oggetti caricati nel tuo bucket.

Impostazione di Object Ownership su un bucket esistente

È possibile configurare S3 Object Ownership su un bucket S3 esistente. Per applicare Object Ownership quando si crea un bucket, consulta Impostazione di Object Ownership quando si crea un bucket.

S3 Object Ownership è un'impostazione a livello di bucket di Amazon S3 che puoi utilizzare per disabilitare le liste di controllo accessi (ACL) e assumere la proprietà di ogni oggetto nel tuo bucket, semplificando la gestione degli accessi per i dati archiviati in Amazon S3. Per impostazione predefinita, Proprietà dell'oggetto S3 è impostata su Proprietario del bucket applicato e le ACL sono disabilitate per nuovi bucket. Con le ACL disabilitate, il proprietario del bucket possiede ogni oggetto nel bucket e gestisce l'accesso ai dati in maniera esclusiva utilizzando policy di gestione dell'accesso. È consigliabile mantenere le ACL disabilitate, tranne in circostanze insolite in cui è necessario controllare individualmente l'accesso per ciascun oggetto.

Object Ownership ha tre impostazioni che è possibile utilizzare per controllare la proprietà degli oggetti caricati nel bucket e disabilitare o abilitare le ACL:

ACL disabilitate

- Proprietario del bucket applicato (impostazione predefinita): le ACL sono disabilitate e il proprietario del bucket possiede automaticamente e ha il controllo completo di ogni oggetto nel bucket. Le ACL non influiscono più sulle autorizzazioni per i dati nel bucket S3. Il bucket utilizza le policy per definire il controllo degli accessi.

ACL abilitate

- Proprietario del bucket scelto – Il proprietario del bucket possiede e ha il pieno controllo sui nuovi oggetti che altri account scrivono nel bucket con l'ACL predefinita `bucket-owner-full-control`.
- Object writer: chi carica un oggetto possiede l'oggetto, ne ha il pieno controllo e può concedere ad altri utenti l'accesso ad esso tramite ACL. Account AWS

Prerequisiti: prima di applicare l'impostazione Proprietario del bucket applicato per disabilitare le ACL, è necessario migrare le autorizzazioni ACL bucket alle policy del bucket e ripristinare le ACL bucket sull'ACL privata di default. Si consiglia inoltre di migrare le autorizzazioni ACL di oggetti alle policy di bucket e di modificare le policy di bucket che richiedono ACL diverse dalle ACL di controllo completo

del proprietario del bucket. Per ulteriori informazioni, consulta Prerequisiti per la disabilitazione delle ACL.

Autorizzazioni: Per utilizzare questa operazione, è necessario disporre dell'autorizzazione `s3:PutBucketOwnershipControls`. Per ulteriori informazioni sulle autorizzazioni di Amazon S3, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket scegliere il nome del bucket al quale applicare un'impostazione S3 Object Ownership.
3. Scegli la scheda Autorizzazioni.
4. Alla voce Proprietà Oggetto scegli Modifica.
5. Alla voce Proprietà oggetto, per disabilitare o abilitare le ACL e controllare la proprietà degli oggetti caricati nel bucket, scegliere una delle seguenti impostazioni:

   ACL disabilitate

   - Proprietario del bucket applicato – Le ACL sono disabilitate e il proprietario del bucket possiede automaticamente e ha il pieno controllo su ogni oggetto nel bucket. Le ACL non influiscono più sulle autorizzazioni per i dati nel bucket S3. Il bucket utilizza le policy per definire il controllo degli accessi.

     Per richiedere che tutti i nuovi bucket vengano creati con gli ACL disattivati utilizzando IAM o AWS Organizations le policy, consulta. Disabilitazione degli ACL per tutti i nuovi bucket (proprietario del bucket applicato)

   ACL abilitate

   - Proprietario del bucket scelto – Il proprietario del bucket possiede e ha il pieno controllo sui nuovi oggetti che altri account scrivono nel bucket con l'ACL predefinita `bucket-owner-full-control`.

     Se applichi l'impostazione del proprietario del bucket preferito per richiedere che tutti i caricamenti di Amazon S3 includano l'ACL predefinita `bucket-owner-full-control`, puoi

aggiungere una policy del bucket che consenta solo il caricamento di oggetti che utilizzano questo ACL.

- Object writer: chi carica un oggetto possiede l'oggetto, ne ha il pieno controllo e può concedere ad altri utenti l'accesso ad esso tramite ACL. Account AWS

6. Selezionare Salva.

Utilizzando il AWS CLI

Per applicare un'impostazione Object Ownership per un bucket esistente, utilizzare il comando `put-bucket-ownership-controls` con il parametro `--ownership-controls`. I valori validi per la proprietà sono`BucketOwnerEnforced`, `BucketOwnerPreferred` o `ObjectWriter`.

In questo esempio viene applicata l'impostazione Proprietario del bucket applicato per un bucket esistente utilizzando la AWS CLI:

```
aws s3api put-bucket-ownership-controls --bucket DOC-EXAMPLE-BUCKET --ownership-
controls="Rules=[{ObjectOwnership=BucketOwnerEnforced}]"
```

Per ulteriori informazioni su `put-bucket-ownership-controls`, consulta put-bucket-ownership-controls nella Guida per l'utente di AWS Command Line Interface .

Utilizzo dell' AWS SDK for Java

In questo esempio viene eseguita l'impostazione `BucketOwnerEnforced` per Object Ownership su un bucket esistente utilizzando la 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);
```

Usando AWS CloudFormation

Per utilizzarlo AWS CloudFormation per applicare un'impostazione di proprietà dell'oggetto a un bucket esistente, consulta AWS::S3::Bucket OwnershipControlsla Guida per l'AWS CloudFormation utente.

Utilizzo di REST API

Per utilizzare REST API per applicare un'impostazione Object Ownership a un bucket S3 esistente, utilizzare `PutBucketOwnershipControls`. Per ulteriori informazioni, consulta PutBucketOwnershipControls in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Fasi successive: dopo aver eseguito le impostazioni Proprietario del bucket applicato o Proprietario del bucket preferito per Proprietà dell'oggetto, è possibile compiere i seguenti passaggi:

- Proprietario del bucket applicato – Richiedi che tutti i nuovi bucket vengano creati con ACL disabilitate utilizzando le policy IAM o di Organizations.
- Proprietario del bucket preferito – Aggiungi una policy di bucket S3 per richiedere l'ACL predefinita `bucket-owner-full-control` per tutti gli oggetti caricati nel tuo bucket.

Visualizzare l'impostazione di Object Ownership per un bucket S3

S3 Object Ownership è un'impostazione a livello di bucket di Amazon S3 che puoi utilizzare per disabilitare le liste di controllo accessi (ACL) e assumere la proprietà di ogni oggetto nel tuo bucket, semplificando la gestione degli accessi per i dati archiviati in Amazon S3. Per impostazione predefinita, Proprietà dell'oggetto S3 è impostata su Proprietario del bucket applicato e le ACL sono disabilitate per nuovi bucket. Con le ACL disabilitate, il proprietario del bucket possiede ogni oggetto nel bucket e gestisce l'accesso ai dati in maniera esclusiva utilizzando policy di gestione dell'accesso. È consigliabile mantenere le ACL disabilitate, tranne in circostanze insolite in cui è necessario controllare individualmente l'accesso per ciascun oggetto.

Object Ownership ha tre impostazioni che è possibile utilizzare per controllare la proprietà degli oggetti caricati nel bucket e disabilitare o abilitare le ACL:

ACL disabilitate

- Proprietario del bucket applicato (impostazione predefinita): le ACL sono disabilitate e il proprietario del bucket possiede automaticamente e ha il controllo completo di ogni oggetto nel bucket. Le ACL non influiscono più sulle autorizzazioni per i dati nel bucket S3. Il bucket utilizza le policy per definire il controllo degli accessi.

ACL abilitate

- Proprietario del bucket scelto – Il proprietario del bucket possiede e ha il pieno controllo sui nuovi oggetti che altri account scrivono nel bucket con l'ACL predefinita `bucket-owner-full-control`.
- Object writer: chi carica un oggetto possiede l'oggetto, ne ha il pieno controllo e può concedere ad altri utenti l'accesso ad esso tramite ACL. Account AWS

È possibile visualizzare le impostazioni di S3 Object Ownership per un bucket Amazon S3. Per impostare Object Ownership per un nuovo bucket, consultare Impostazione di Object Ownership quando si crea un bucket. Per impostare Object Ownership per un bucket esistente, consultare Impostazione di Object Ownership su un bucket esistente.

Autorizzazioni: Per utilizzare questa operazione, è necessario disporre dell'autorizzazione `s3:GetBucketOwnershipControls`. Per ulteriori informazioni sulle autorizzazioni di Amazon S3, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket scegliere il nome del bucket al quale applicare un'impostazione di Object Ownership.
3. Scegli la scheda Autorizzazioni.
4. Alla voce Proprietà Oggetto, è possibile visualizzare le impostazioni di Object Ownership per il bucket.

Usando il AWS CLI

Per recuperare l'impostazione S3 Object Ownership per un bucket S3, usa il comando. get-bucket-ownership-controls AWS CLI

```
aws s3api get-bucket-ownership-controls --bucket DOC-EXAMPLE-BUCKET
```

Utilizzo di REST API

Per recuperare l'impostazione di Object Ownership per un bucket S3, utilizzare l'operazione API `GetBucketOwnershipControls`. Per ulteriori informazioni, consulta GetBucketOwnershipControls.

Disabilitare le ACL per tutti i nuovi bucket e applicare Object Ownership

Ti consigliamo di disabilitare le ACL sui bucket Amazon S3. È possibile farlo applicando l'impostazione Proprietario del bucket applicato per S3 Proprietà dell'oggetto. Una volta applicata questa impostazione, le ACL verranno disabilitate e si possiederanno automaticamente e si avrà il pieno controllo su tutti gli oggetti del bucket. Per richiedere che tutti i nuovi bucket vengano creati con gli ACL disattivati, utilizza le politiche AWS Identity and Access Management (IAM) o le politiche di controllo dei AWS Organizations servizi (SCP), come descritto nella sezione successiva.

Per applicare la proprietà degli oggetti per i nuovi oggetti senza disabilitare le ACL, è possibile eseguire l'impostazione proprietario del bucket preferito. Una volta applicata questa impostazione, si consiglia fortemente di aggiornare la policy del bucket per richiedere l'ACL predefinita`bucket-owner-full-control` per tutte le richieste PUT sul tuo bucket. I client devono anch'essi essere aggiornati per inviare l'ACL predefinita `bucket-owner-full-control` al tuo bucket da altri account.

Argomenti

- Disabilitazione degli ACL per tutti i nuovi bucket (proprietario del bucket applicato)
- Richiedere l'ACL bucket-owner-full-control predefinito per le operazioni di Amazon PUT S3 (preferibilmente il proprietario del bucket)

Disabilitazione degli ACL per tutti i nuovi bucket (proprietario del bucket applicato)

La seguente policy IAM di esempio nega l'autorizzazione `s3:CreateBucket` per un utente IAM o un ruolo specifico a meno che non venga applicata l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto. La coppia chiave-valore nel blocco di `Condition` specifica `s3:x-`

`amz-object-ownership` come chiave e l'impostazione `BucketOwnerEnforced` come valore corrispondente. In altre parole, l'utente IAM può creare bucket solo se imposta Proprietario del bucket applicato per Proprietà dell'oggetto e disabilita le ACL. Puoi anche utilizzare questa policy come SCP limite per la tua organizzazione. AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RequireBucketOwnerFullControl",
            "Action": "s3:CreateBucket",
            "Effect": "Deny",
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-object-ownership": "BucketOwnerEnforced"
                }
            }
        }
    ]
}
```

Richiedere l'ACL bucket-owner-full-control predefinito per le operazioni di Amazon **PUT** S3 (preferibilmente il proprietario del bucket)

Con l'impostazione proprietario del bucket preferito per Object Ownership, in qualità di proprietario del bucket possiedi e hai il pieno controllo sui nuovi oggetti che gli altri account scrivono sul tuo bucket con l'ACL predefinita `bucket-owner-full-control`. Tuttavia, se altri account scrivono oggetti nel tuo bucket senza l'ACL predefinita `bucket-owner-full-control`, l'object writer mantiene il pieno controllo degli accessi. In qualità di proprietario del bucket, è possibile implementare una policy del bucket che consenta la scrittura solo se si specifica l'ACL predefinita `bucket-owner-full-control`.

> **Note**
>
> Se le ACL sono disabilitate con l'impostazione Proprietario del bucket applicato, in qualità di proprietario del bucket possiedi automaticamente e hai il controllo completo di tutti gli oggetti del bucket. Non è necessario utilizzare questa sezione per aggiornare la policy del bucket per applicare la proprietà degli oggetti per il proprietario del bucket.

La seguente policy del bucket specifica che l'account *111122223333* può caricare oggetti *DOC-EXAMPLE-BUCKET* solo quando l'ACL dell'oggetto è impostata su `bucket-owner-full-control`. Assicurati di sostituire *111122223333* con un account reale e *DOC-EXAMPLE-BUCKET* con il nome del tuo 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:::DOC-EXAMPLE-BUCKET/*",
         "Condition": {
            "StringEquals": {
               "s3:x-amz-acl": "bucket-owner-full-control"
            }
         }
      }
   ]
}
```

Di seguito è riportata un'operazione di copia di esempio che include l'ACL `bucket-owner-full-control` predefinito utilizzando AWS Command Line Interface (AWS CLI).

```
aws s3 cp file.txt s3://DOC-EXAMPLE-BUCKET --acl bucket-owner-full-control
```

Dopo che la policy del bucket è diventata efficace, se il client non include l'ACL predefinita `bucket-owner-full-control`, l'operazione non riuscirà e l'uploader riceverà il seguente errore:

Si è verificato un errore (AccessDenied) durante la chiamata dell' PutObject operazione: Accesso negato.

> **Note**
>
> Se i client hanno bisogno di accedere agli oggetti dopo il caricamento, sarà necessario concedere autorizzazioni aggiuntive per l'account di caricamento. Per informazioni sulla concessione agli account dell'accesso alle risorse, consulta la sezione Procedure dettagliate che utilizzano policy per gestire l'accesso alle risorse Amazon S3.

Risoluzione dei problemi

Quando esegui l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto S3, le liste di controllo degli accessi (ACL) vengono disabilitate e tu, in qualità di proprietario del bucket, possiedi automaticamente tutti gli oggetti nel bucket. Le ACL non influiscono più sulle autorizzazioni per gli oggetti nel bucket. Puoi utilizzare le policy per concedere autorizzazioni. Tutte le richieste PUT S3 devono specificare l'ACL predefinita `bucket-owner-full-control` o non specificare una ACL; in caso contrario non andranno a buon fine. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

Se viene specificato un'ACL non valida o le autorizzazioni ACL del bucket garantiscono l'accesso al di fuori del tuo Account AWS, potrebbero essere visualizzate le seguenti risposte di errore.

AccessControlListNotSupported

Dopo aver applicato l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto, le ACL vengono disabilitate. Le richieste di impostazione degli ACL o di aggiornamento degli ACL hanno esito negativo e restituiscono il codice AccessControlListNotSupporteddi errore. `400` Le richieste di lettura delle ACL sono ancora supportate. Le richieste di lettura delle ACL restituiscono sempre una risposta che mostra il pieno controllo per il proprietario del bucket. Nelle operazioni PUT è necessario specificare le ACL di controllo completo del proprietario del bucket o non specificare un'ACL. Altrimenti, le tue operazioni PUT falliranno.

Il `put-object` AWS CLI comando di esempio seguente include l'`public-read`ACL predefinito.

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key object-key-name --body doc-example-body --acl public-read
```

Se il bucket utilizza l'impostazione Proprietario del bucket applicato per disabilitare le ACL, questa operazione non va a buon fine e l'uploader riceve il seguente messaggio di errore:

Si è verificato un errore (AccessControlListNotSupported) durante la chiamata dell'PutObject operazione: il bucket non consente gli ACL

InvalidBucketAclWithObjectOwnership

Se desideri applicare l'impostazione Proprietario del bucket applicato per disabilitare le ACL, l'ACL bucket dovrà fornire il controllo completo solo al proprietario del bucket. L'ACL del bucket non può consentire l'accesso a un gruppo esterno Account AWS o a qualsiasi altro gruppo. Ad esempio, se la tua `CreateBucket` richiesta imposta Bucket owner enforced e specifica un bucket ACL che fornisce l'accesso a un bucket ACL che fornisce l'accesso a un file esterno Account AWS, la richiesta ha esito negativo e restituisce il codice di errore. `400` InvalidBucketAclWithObjectOwnership Allo stesso modo, se la tua richiesta `PutBucketOwnershipControls` imposta il proprietario del bucket applicato su un bucket con un ACL di bucket che concede autorizzazioni ad altri, la richiesta avrà esito negativo.

Example : l'ACL del bucket esistente concede l'accesso pubblico in lettura

Ad esempio, se un ACL bucket esistente concede l'accesso pubblico in lettura, non potrai applicare l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto finché queste autorizzazioni ACL non vengono migrate a una policy del bucket e l'ACL bucket non viene ripristinata sull'ACL privata di default. Per ulteriori informazioni, consulta Prerequisiti per la disabilitazione delle ACL.

Questo esempio di ACL di bucket concede l'accesso pubblico in lettura:

```
{
    "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"
        }
    ]
}
```

Il `put-bucket-ownership-controls` AWS CLI comando di esempio seguente applica l'impostazione Bucket owner enforced per Object Ownership:

```
aws s3api put-bucket-ownership-controls --bucket DOC-EXAMPLE-BUCKET --ownership-
controls Rules=[{ObjectOwnership=BucketOwnerEnforced}]
```

Poiché l'ACL del bucket consente l'accesso pubblico in lettura, la richiesta sarà respinta e restituirà il seguente codice di errore:

Si è verificato un errore (InvalidBucketAclWithObjectOwnership) durante la chiamata dell' PutBucketOwnershipControls operazione: Bucket non può avere ACL impostati con l'impostazione ObjectOwnership BucketOwnerEnforced

Utilizzo della funzionalità Cross-Origin Resource Sharing (CORS)

La funzionalità CORS (Cross-Origin Resource Sharing, condivisione delle risorse multiorigine) definisce un metodo con cui le applicazioni Web dei clienti caricate in un dominio possono interagire con le risorse situate in un dominio differente. Con il supporto della funzionalità CORS, è possibile creare applicazioni Web lato client complete con Amazon S3 e concedere l'accesso multiorigine alle risorse di Amazon S3 in modo selettivo.

In questa sezione viene fornita una panoramica della funzionalità CORS. Gli argomenti secondari descrivono come abilitare CORS utilizzando la console Amazon S3 o a livello di codice utilizzando l'API REST di Amazon S3 e gli SDK. AWS

Cross Origin Resource Sharing (CORS): scenari dei casi d'uso

Di seguito sono riportati alcuni scenari di esempio per l'uso della funzionalità CORS.

Scenario 1

Si supponga di ospitare un sito Web in un bucket Amazon S3 denominato `website`, come descritto in Hosting di un sito Web statico tramite Amazon S3. Gli utenti caricano l'endpoint del sito Web:

```
http://website.s3-website.us-east-1.amazonaws.com
```

Ora vuoi utilizzarlo JavaScript sulle pagine Web archiviate in questo bucket per poter effettuare richieste GET e PUT autenticate sullo stesso bucket utilizzando l'endpoint dell'API Amazon S3 per il bucket,. `website.s3.us-east-1.amazonaws.com` Normalmente un browser JavaScript impedirebbe l'autorizzazione di tali richieste, ma con CORS puoi configurare il tuo bucket per abilitare esplicitamente le richieste provenienti da più origini. `website.s3-website.us-east-1.amazonaws.com`

Scenario 2

Si supponga di voler ospitare un font Web dal bucket S3. Anche in questo caso, i browser richiedono un controllo della funzionalità CORS (anche denominato "controllo preliminare") per il caricamento dei font Web. È necessario configurare il bucket che ospita il font Web in modo da consentire a qualsiasi origine di eseguire queste richieste.

In che modo Amazon S3 valuta la configurazione CORS in un bucket?

Quando Amazon S3 riceve una richiesta preliminare da un browser, valuta la configurazione CORS per il bucket e utilizza la prima regola `CORSRule` corrispondente alla richiesta del browser in entrata per abilitare una richiesta multiorigine. Per garantire la corrispondenza tra la regola e la richiesta, è necessario che siano soddisfatte le condizioni elencate di seguito.

- L'intestazione `Origin` della richiesta deve corrispondere a un elemento `AllowedOrigin`.
- Il metodo della richiesta (ad esempio, GET o PUT) o l'intestazione `Access-Control-Request-Method` in caso di richiesta `OPTIONS` preliminare deve essere uno degli elementi `AllowedMethod`.
- Ogni intestazione elencata nell'intestazione `Access-Control-Request-Headers` nella richiesta preliminare deve corrispondere a un elemento `AllowedHeader`.

> **Note**
>
> Le ACL e le policy continuano a essere valide quando si abilita la funzionalità CORS nel bucket.

In che modo Punto di accesso per le espressioni Lambda dell'oggetto supporta CORS

Quando Lambda per oggetti Amazon S3 riceve una richiesta da un browser o la richiesta include un'intestazione `Origin`, Lambda per oggetti Amazon S3 aggiunge sempre un campo di intestazione `“AllowedOrigins":"*"`.

Per ulteriori informazioni sull'uso di CORS, consulta gli argomenti riportati di seguito.

Argomenti

- Configurazione CORS
- Configurazione della funzionalità Cross-Origin Resource Sharing (CORS)

Configurazione CORS

Per configurare il bucket in modo da consentire le richieste multiorigine, si crea una configurazione CORS. La configurazione CORS è un documento XML con le regole che identificano le origini che potranno accedere al bucket, le operazioni (metodi HTTP) supportate per ogni origine e altre informazioni specifiche dell'operazione. È possibile aggiungere fino a 100 regole alla configurazione. È possibile aggiungere la configurazione CORS come risorsa secondaria `cors` al bucket.

Se configura CORS nella console S3, è necessario utilizzare JSON per creare una configurazione CORS. La nuova console S3 supporta solo configurazioni JSON CORS.

Per ulteriori informazioni sulla configurazione CORS e sugli elementi in essa contenuti, consulta gli argomenti riportati di seguito. Per istruzioni su come aggiungere una configurazione CORS, consulta Configurazione della funzionalità Cross-Origin Resource Sharing (CORS).

> **Important**
>
> Nella console S3, la configurazione CORS deve essere JSON.

Argomenti

- Esempio 1
- Esempio 2
- AllowedMethod elemento

- AllowedOrigin elemento
- AllowedHeader elemento
- ExposeHeader elemento
- MaxAgeSeconds elemento

Esempio 1

Anziché accedere a un sito Web utilizzando un endpoint del sito Web Amazon S3, è possibile utilizzare il proprio dominio, come `example1.com`, per consegnare il contenuto. Per informazioni sull'uso del proprio dominio, consulta Tutorial: Configurazione di un sito Web statico utilizzando un dominio personalizzato registrato con Route 53.

La configurazione `cors` di esempio riportata di seguito include tre regole, specificate come elementi `CORSRule`:

- La prima regola consente le richieste multiorigine PUT, POST e DELETE provenienti dall'origine `http://www.example1.com` La regole consente inoltre tutte le intestazioni in una richiesta OPTIONS preliminare tramite l'intestazione `Access-Control-Request-Headers`. In risposta alle richieste OPTIONS preliminari, Amazon S3 restituisce le intestazioni richieste.
- La seconda regola consente le stesse richieste multiorigine della prima regola, ma si applica a un'altra origine, `http://www.example2.com`.
- La terza regola consente le richieste multiorigine GET provenienti da tutte le origini. Il carattere jolly `*` si riferisce a tutte le origini.

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>
```

Esempio 2

La configurazione CORS supporta anche i parametri di configurazione opzionali, come illustrato nella seguente configurazione CORS. In questo esempio la configurazione CORS consente le richieste multiorigine PUT, POST e DELETE provenienti dall'origine `http://www.example.com`.

JSON

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "http://www.example.com"
        ],
        "ExposeHeaders": [
            "x-amz-server-side-encryption",
            "x-amz-request-id",
            "x-amz-id-2"
        ],
        "MaxAgeSeconds": 3000
    }
```

```
]
```

XML

```
<CORSConfiguration>
 <CORSRule>
   <AllowedOrigin>http://www.example.com</AllowedOrigin>
   <AllowedMethod>PUT</AllowedMethod>
   <AllowedMethod>POST</AllowedMethod>
   <AllowedMethod>DELETE</AllowedMethod>
   <AllowedHeader>*</AllowedHeader>
  <MaxAgeSeconds>3000</MaxAgeSeconds>
  <ExposeHeader>x-amz-server-side-encryption</
ExposeHeader>
  <ExposeHeader>x-amz-request-id</
ExposeHeader>
  <ExposeHeader>x-amz-id-2</ExposeHeader>
 </CORSRule>
</CORSConfiguration>
```

L'elemento `CORSRule` nella configurazione precedente include gli elementi opzionali riportati di seguito.

- `MaxAgeSeconds` – Specifica l'intervallo di tempo in secondi (in questo esempio, 3000) durante il quale il browser memorizza nella cache una risposta Amazon S3 a una richiesta OPTIONS preliminare per la risorsa specificata. La memorizzazione nella cache della risposta consente al browser di non inviare richieste preliminari ad Amazon S3 se la richiesta originale viene ripetuta.
- `ExposeHeader`—Identifica le intestazioni di risposta (in questo esempio, `x-amz-server-side-encryption`x-amz-request-id, `ex-amz-id-2`) a cui i clienti possono accedere dalle loro applicazioni (ad esempio, da un oggetto). JavaScript `XMLHttpRequest`

AllowedMethod elemento

Nella configurazione CORS è possibile specificare i valori indicati di seguito per l'elemento `AllowedMethod`.

- GET
- PUT
- POST

- DELETE
- HEAD

AllowedOrigin elemento

Nell'elemento `AllowedOrigin`, è possibile specificare le origini da cui si desiderano consentire le richieste multidominio, ad esempio `http://www.example.com`. La stringa di origine può contenere solamente un carattere jolly `*`, ad esempio `http://*.example.com`. Se si desidera, è possibile specificare `*` come origine per consentire a tutte le origini di inviare richieste multiorigine. È anche possibile specificare `https` per abilitare solo le origini sicure.

AllowedHeader elemento

L'elemento `AllowedHeader` specifica le intestazioni consentite in una richiesta preliminare tramite l'intestazione `Access-Control-Request-Headers`. Ogni nome di intestazione in `Access-Control-Request-Headers` deve corrispondere a una voce nella regola. Tra le intestazioni richieste, Amazon S3 invierà nella risposta solo quelle consentite. Per un esempio di elenco di intestazioni che possono essere utilizzate nelle richieste ad Amazon S3, consulta l'argomento relativo alle intestazioni di richiesta comuni nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Ogni AllowedHeader stringa nella regola può contenere al massimo un carattere jolly (*). Ad esempio, `<AllowedHeader>x-amz-*</AllowedHeader>` abiliterà tutte le intestazioni specifiche di Amazon.

ExposeHeader elemento

Ogni `ExposeHeader` elemento identifica un'intestazione nella risposta a cui desideri che i clienti possano accedere dalle loro applicazioni (ad esempio, da un JavaScript `XMLHttpRequest` oggetto). Per un elenco delle intestazioni di risposta più comuni di Amazon S3, consulta l'argomento relativo alle intestazioni di richiesta comuni nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

MaxAgeSeconds elemento

L'elemento `MaxAgeSeconds` specifica l'intervallo di tempo in secondi durante il quale il browser può memorizzare nella cache la risposta a una richiesta preliminare identificata in base a risorsa, metodo HTTP e origine.

Configurazione della funzionalità Cross-Origin Resource Sharing (CORS)

La funzionalità CORS (Cross-Origin Resource Sharing, condivisione delle risorse multiorigine) definisce un metodo con cui le applicazioni Web dei clienti caricate in un dominio possono interagire con le risorse situate in un dominio differente. Con il supporto della funzionalità CORS, è possibile creare applicazioni Web lato client complete con Amazon S3 e concedere l'accesso multiorigine alle risorse di Amazon S3 in modo selettivo.

Questa sezione mostra come abilitare CORS utilizzando la console Amazon S3, l'API REST di Amazon S3 e gli SDK. AWS Per configurare il bucket in modo da consentire richieste tra più origini, è necessario aggiungere una configurazione CORS al bucket. La configurazione CORS è un documento in cui sono definite regole che identificano le origini che potranno accedere al bucket, le operazioni (metodi HTTP) supportate per ogni origine e altre informazioni specifiche dell'operazione. Nella console S3, la configurazione CORS deve essere un documento JSON.

Per esempi di configurazioni CORS in JSON e XML, consulta Configurazione CORS.

Utilizzo della console S3

In questa sezione viene descritto come utilizzare la console di Amazon S3 per aggiungere una configurazione CORS (Cross-Origin Resource Sharing, condivisione delle risorse multiorigine) a un bucket S3.

Quando si abilita la funzionalità CORS nel bucket, le liste di controllo degli accessi (ACL) e altre policy di autorizzazione di accesso continuano ad essere valide.

> **Important**
>
> Nella nuova console S3, la configurazione CORS deve essere JSON. Per esempi di configurazioni CORS in JSON e XML, consulta Configurazione CORS.

Per aggiungere una configurazione CORS a un bucket S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket name (Nome bucket) scegliere il nome del bucket per il quale si desidera creare una policy di bucket.
3. Seleziona Autorizzazioni.

4. Nella sezione Cross-Origin Resource Sharing (CORS) scegliere Edit (Modifica).
5. Nella casella di testo CORS configuration editor (Editor configurazione CORS), digitare o copiare e incollare una nuova configurazione CORS oppure modificare una configurazione esistente.

   La configurazione CORS è un file JSON. Il testo digitato nell'editor deve essere in formato JSON valido. Per ulteriori informazioni, consulta Configurazione CORS.
6. Seleziona Salva modifiche.

> **Note**
>
> Amazon S3 visualizza l'Amazon Resource Name (ARN) per il bucket accanto al titolo CORS configuration editor (Editor configurazione CORS). Per ulteriori informazioni sugli ARN, consulta Amazon Resource Names (ARNs) e AWS Service Namespaces nel. Riferimenti generali di Amazon Web Services

Utilizzo degli SDK AWS

Puoi utilizzare l' AWS SDK per gestire la condivisione di risorse tra origini diverse (CORS) per un bucket. Per ulteriori informazioni sulla funzionalità CORS, consulta Utilizzo della funzionalità Cross-Origin Resource Sharing (CORS).

Gli esempi seguenti:

- Crea una configurazione CORS e imposta la configurazione su un bucket
- Recupera la configurazione e la modifica aggiungendo una regola
- Aggiunge la configurazione modificata al bucket
- Elimina la configurazione

Java

Example

Example

Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella Developer Guide. 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

Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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()));
            }
        }
    }
}
```

Utilizzo di REST API

Per impostare una configurazione CORS nel bucket, è possibile utilizzare la AWS Management Console. Se l'applicazione lo richiede, si può inoltre inviare le richieste REST direttamente. Le sezioni seguenti della Documentazione d riferimento delle API di Amazon Simple Storage Service descrivono le operazioni di REST API correlate alla configurazione CORS.

- PutBucketCors
- GetBucketCors
- DeleteBucketCors
- OPTIONS object

Registrazione e monitoraggio in Amazon S3

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di Amazon S3 e delle tue AWS soluzioni. È necessario raccogliere i dati di monitoraggio da tutte le parti della AWS soluzione in modo da poter eseguire più facilmente il debug di un errore multipunto, se si verifica. AWS fornisce diversi strumenti per monitorare le risorse Amazon S3 e rispondere a potenziali incidenti.

Per ulteriori informazioni, consulta Monitoraggio di Amazon S3.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

CloudWatch Allarmi Amazon

Utilizzando Amazon CloudWatch alarms, controlli una singola metrica per un periodo di tempo specificato. Se la metrica supera una determinata soglia, viene inviata una notifica a un argomento o una policy di Amazon SNS. AWS Auto Scaling CloudWatch gli allarmi non richiamano azioni perché si trovano in uno stato particolare. È necessario invece cambiare lo stato e mantenerlo per un numero di periodi specificato. Per ulteriori informazioni, consulta Monitoraggio delle metriche con Amazon CloudWatch.

AWS CloudTrail Registri

CloudTrail fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in Amazon S3. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare la richiesta effettuata ad Amazon S3, l'indirizzo IP da cui è stata effettuata, chi ha effettuato la richiesta, quando è stata effettuata e ulteriori dettagli. Per ulteriori informazioni, consulta Registrazione delle chiamate API Amazon S3 tramite AWS CloudTrail.

Amazon GuardDuty

Amazon GuardDuty è un servizio di rilevamento delle minacce che monitora continuamente account, contenitori, carichi di lavoro e dati all'interno del tuo AWS ambiente per identificare potenziali minacce o rischi per la sicurezza dei tuoi bucket S3. GuardDuty fornisce inoltre un contesto completo sulle minacce che rileva. GuardDuty monitora i registri AWS CloudTrail di gestione alla ricerca di minacce e visualizza le informazioni rilevanti per la sicurezza. Ad esempio,

GuardDuty includerà fattori di una richiesta API, come l'utente che ha effettuato la richiesta, la posizione da cui è stata effettuata la richiesta e l'API specifica richiesta, che potrebbe essere insolita nel tuo ambiente. GuardDuty S3 Protection monitora gli eventi relativi ai dati S3 raccolti CloudTrail e identifica comportamenti potenzialmente anomali e dannosi in tutti i bucket S3 dell'ambiente.

Log di accesso Amazon S3

I log di accesso al server forniscono record dettagliati relativi alle richieste che vengono effettuate a un bucket. I log di accesso al server sono utili per numerose applicazioni. Ad esempio, le informazioni del log di accesso possono essere utili nei controlli di accesso e di sicurezza. Per ulteriori informazioni, consulta Registrazione delle richieste con registrazione dell'accesso al server.

AWS Trusted Advisor

Trusted Advisor si basa sulle migliori pratiche apprese servendo centinaia di migliaia di clienti. AWS Trusted Advisor ispeziona l' AWS ambiente e quindi formula raccomandazioni quando esistono opportunità per risparmiare denaro, migliorare la disponibilità e le prestazioni del sistema o contribuire a colmare le lacune di sicurezza. Tutti i AWS clienti hanno accesso a cinque Trusted Advisor controlli. I clienti con un piano di supporto Business o Enterprise possono visualizzare tutti i Trusted Advisor controlli.

Trusted Advisor dispone dei seguenti controlli relativi ad Amazon S3:

- Registrazione della configurazione dei bucket Amazon S3.
- Controlli della sicurezza per i bucket di Amazon S3 dotati di autorizzazioni di accesso aperte.
- Controlli della tolleranza ai guasti per i bucket di Amazon S3 per i quali la funzione Controllo delle versioni non è abilitata o è sospesa.

Per ulteriori informazioni, consulta AWS Trusted Advisor nella Guida per l'utente di AWS Support .

Le best practice di sicurezza seguenti gestiscono anche il logging e il monitoraggio:

- Identify and audit all your Amazon S3 buckets
- Implement monitoring using Amazon Web Services monitoring tools
- Attiva AWS Config
- Enable Amazon S3 server access logging
- Use CloudTrail

- Monitor Amazon Web Services security advisories

Convalida della conformità per Amazon S3

La sicurezza e la conformità di Amazon S3 vengono valutate da revisori di terze parti nell'ambito di diversi programmi di AWS conformità, tra cui:

- System and Organization Controls (SOC)
- Payment Card Industry Data Security Standard (PCI DSS)
- Federal Risk and Authorization Management Program (FedRAMP)
- Health Insurance Portability and Accountability Act (HIPAA)

AWS fornisce un elenco di AWS servizi aggiornato di frequente nell'ambito di specifici programmi di conformità nella pagina AWS Services in Scope by Compliance Program.

I report di audit di terze parti possono essere scaricati utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact.

Per ulteriori informazioni sui programmi di AWS conformità, consulta Programmi di AWS conformità.

La responsabilità della conformità durante l'utilizzo di Amazon S3 è determinata dalla riservatezza dei dati, dagli obiettivi di conformità dell'organizzazione e dalle leggi e normative in vigore. Se l'utilizzo di Amazon S3 è soggetto alla conformità a standard come HIPAA, PCI o FedRAMP, AWS fornisce alcune risorse utili:

- Guide introduttive su sicurezza e conformità che illustrano le considerazioni sull'architettura e i passaggi per implementare ambienti di base incentrati sulla sicurezza e la conformità. AWS
- Architecting for HIPAA Security and Compliance descrive in che modo le aziende utilizzano per aiutarle a soddisfare i requisiti HIPAA. AWS
- AWS Le risorse per la conformità forniscono diverse cartelle di lavoro e guide che potrebbero essere applicabili al settore e alla località in cui operate.
- AWS Config è utile per valutare il livello di conformità delle configurazioni delle risorse con pratiche interne, linee guida e regolamenti del settore.
- AWS Security Hub ti offre una visione completa del tuo stato di sicurezza interno AWS e ti aiuta a verificare la tua conformità agli standard e alle migliori pratiche del settore della sicurezza.
- Utilizzo del blocco oggetti S3 consente di soddisfare i requisiti tecnici degli organi di regolamentazione finanziaria (ad esempio, SEC, FINRA e CFTC) che richiedono storage dei dati WORM (Write Once, Read Many) per alcuni tipi di informazioni su registri e record.

- Amazon S3 Inventory permette di svolgere revisioni e creare report sullo stato di replica e crittografia degli oggetti per esigenze aziendali, normative e di conformità.

Resilienza in Amazon S3

L'infrastruttura AWS globale è costruita attorno a regioni e zone di disponibilità. Regioni AWS forniscono zone di disponibilità multiple, fisicamente separate e isolate, collegate con reti a bassa latenza, throughput elevato e altamente ridondante. Queste zone di disponibilità offrono un modo efficace per progettare e gestire le applicazioni e i database. Sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali. Se avete specificamente bisogno di replicare i dati su distanze geografiche maggiori, potete utilizzarePanoramica sulla replica degli oggetti, che consente la copia automatica e asincrona degli oggetti tra bucket diversi. Regioni AWS

Ciascuno ha più zone di disponibilità. Regione AWS Puoi distribuire le applicazioni tra più zone di disponibilità nella stessa regione per avere maggiore tolleranza ai guasti e una bassa latenza. Le zone di disponibilità sono collegate tra loro con velocissime reti in fibra ottica private, per consentire ai clienti di progettare applicazioni che eseguano il failover su diverse zone di disponibilità senza provocare interruzioni.

Per ulteriori informazioni sulle zone Regioni AWS di disponibilità, vedere AWS Global Infrastructure.

Oltre all'infrastruttura AWS globale, Amazon S3 offre diverse funzionalità per supportare le esigenze di resilienza e backup dei dati.

Configurazione del ciclo di vita

Una configurazione del ciclo di vita è un insieme di regole che definiscono le operazioni applicate da Amazon S3 a un gruppo di oggetti. Tramite le regole di configurazione del ciclo di vita, è possibile indicare ad Amazon S3 di trasferire gli oggetti in classi di storage meno costose, archiviarli o eliminarli. Per ulteriori informazioni, consulta Gestione del ciclo di vita dello storage.

Funzione Controllo delle versioni

La funzione Controllo delle versioni è un modo per conservare più versioni di un oggetto nello stesso bucket. La funzione Controllo delle versioni può essere impiegata per conservare, recuperare e ripristinare qualsiasi versione di ogni oggetto archiviato nel bucket Amazon S3. Con la funzione Controllo delle versioni si può facilmente eseguire il ripristino dopo errori dell'applicazione e operazioni non intenzionali dell'utente. Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Blocco di oggetti in S3

Puoi utilizzare il blocco oggetti S3 per archiviare gli oggetti utilizzando il modello write once, read many (WORM). Utilizzando il blocco oggetti S3, puoi impedire che un oggetto venga eliminato o sovrascritto per un determinato periodo di tempo o in modo indefinito. Il blocco oggetti S3 consente di soddisfare i requisiti normativi che richiedono uno storage WORM o semplicemente di aggiungere un ulteriore livello di protezione contro le modifiche e l'eliminazione degli oggetti. Per ulteriori informazioni, consulta Utilizzo del blocco oggetti S3.

Classi di storage

Amazon S3 offre una gamma di classi di archiviazione tra cui scegliere in base ai requisiti del carico di lavoro. Le classi di archiviazione S3 Standard-IA e S3 One Zone-IA sono progettate per i dati a cui si accede almeno una volta al mese e richiedono l'accesso in millisecondi. La classe di archiviazione S3 Glacier Instant Retrieval è progettata per i dati di archiviazione di lunga durata a cui si accede in millisecondi circa una volta al trimestre. Per i dati di archiviazione che non richiedono accesso immediato, come i backup, è possibile utilizzare le classi di archiviazione S3 Glacier Flexier Retrieval o S3 Glacier Deep Archive. Per ulteriori informazioni, consulta Utilizzo delle classi di storage di Amazon S3.

Le best practice di sicurezza seguenti gestiscono anche la resilienza:

- Enable versioning
- Consider Amazon S3 cross-region replication
- Identify and audit all your Amazon S3 buckets

Crittografia dei backup di Amazon S3

Se si esegue l'archiviazione di backup utilizzando Amazon S3, la crittografia dei backup dipende dalla configurazione di tali bucket. Amazon S3 offre un modo per impostare il comportamento di crittografia predefinita per un bucket S3. Puoi configurare la crittografare predefinita di un bucket in modo che gli oggetti siano crittografati quando vengono memorizzati nel bucket. La crittografia predefinita supporta le chiavi archiviate in AWS KMS (SSE-KMS). Per ulteriori informazioni, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3.

Per ulteriori informazioni sulla funzione Controllo delle versioni e sul blocco oggetti, consulta i seguenti argomenti: Utilizzo della funzione Controllo delle versioni nei bucket S3 Utilizzo del blocco oggetti S3

Sicurezza dell'infrastruttura in Amazon S3

In quanto servizio gestito, Amazon S3 è protetto dalle procedure di sicurezza di rete AWS globali descritte nel pilastro di sicurezza del Well-Architected AWS Framework.

L'accesso ad Amazon S3 tramite la rete avviene tramite API AWS pubblicate. I client devono supportare Transport Layer Security (TLS) 1.2. È consigliabile anche il supporto di TLS 1.3. (Per ulteriori informazioni su questa raccomandazione, consulta Connessioni AWS cloud più veloci con TLS 1.3 sul AWS Security Blog.) I client devono inoltre supportare le suite di cifratura con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). Inoltre, le richieste devono essere firmate utilizzando AWS Signature V4 o AWS Signature V2, richiedendo l'immissione di credenziali valide.

Queste API possono essere invocate da qualsiasi posizione di rete. Tuttavia, Amazon S3 supporta anche le policy di accesso basate sulle risorse, che possono includere limitazioni in base all'indirizzo IP di origine. È possibile utilizzare le policy del bucket Amazon S3 per controllare l'accesso ai bucket da endpoint Virtual Private Cloud (VPC) o da VPC specifici. In effetti, questo isola l'accesso alla rete a un determinato bucket Amazon S3 solo dal VPC specifico all'interno della rete. AWS Per ulteriori informazioni, consulta Controllo dell'accesso dagli endpoint VPC con policy di bucket.

Le best practice di sicurezza seguenti gestiscono anche la sicurezza dell'infrastruttura in Amazon S3:

- Consider VPC endpoints for Amazon S3 access
- Identify and audit all your Amazon S3 buckets

Analisi della configurazione e delle vulnerabilità in Amazon S3

AWS gestisce attività di sicurezza di base come l'applicazione di patch al sistema operativo (OS) guest e al database, la configurazione del firewall e il disaster recovery. Queste procedure sono state riviste e certificate dalle terze parti appropriate. Per ulteriori dettagli, consulta le seguenti risorse :

- Convalida della conformità per Amazon S3
- Modello di responsabilità condivisa
- Amazon Web Services: panoramica dei processi di sicurezza

Le best practice di sicurezza seguenti gestiscono anche l'analisi di configurazione e vulnerabilità in Amazon S3:

- Identify and audit all your Amazon S3 buckets
- Attiva AWS Config

Best practice di sicurezza per Amazon S3

Amazon S3 fornisce una serie di caratteristiche di sicurezza che occorre valutare durante lo sviluppo e l'implementazione delle policy di sicurezza. Le seguenti best practice sono linee guida generali e non rappresentano una soluzione di sicurezza completa. Poiché queste best practice potrebbero non essere appropriate o sufficienti per il tuo ambiente, considerale come consigli utili più che prescrizioni.

Argomenti

Best practice di sicurezza per Amazon S3

Le seguenti best practice per Amazon S3 consentono di evitare incidenti di sicurezza.

Disabilitare le liste di controllo degli accessi (ACL)

S3 Proprietà dell'oggetto è un'impostazione a livello di bucket Amazon S3 che è possibile utilizzare per controllare la proprietà degli oggetti caricati nel bucket e per disabilitare o abilitare ACL. Per impostazione predefinita, Proprietà dell'oggetto è impostata su Proprietario del bucket applicato e tutte le ACL sono disabilitate. Quando le ACL sono disabilitate, il proprietario del bucket dispone di tutti gli oggetti nel bucket e gestisce l'accesso ai dati in maniera esclusiva utilizzando policy di gestione dell'accesso.

La maggior parte dei casi d'uso moderno in Amazon S3 non richiede più l'uso di liste di controllo degli accessi (ACL). È consigliabile disabilitare le ACL, tranne in circostanze insolite in cui è necessario controllare individualmente l'accesso per ciascun oggetto. Per disabilitare le ACL e assumere la proprietà di ogni oggetto del tuo bucket, applica l'impostazione Bucket owner enforced (Applicata da proprietario bucket) per S3 Object Ownership. Quando si disabilitano le ACL, è possibile mantenere facilmente un bucket con oggetti caricati da diversi Account AWS.

Quando le ACL sono disabilitate, il controllo degli accessi per i dati è basato su policy, come quelle elencate di seguito:

- AWS Identity and Access Management politiche utente (IAM)
- Policy di bucket S3
- Policy di endpoint del cloud privato virtuale (VPC)
- AWS Organizations politiche di controllo del servizio (SCP)

La disabilitazione delle ACL semplifica la gestione e il controllo delle autorizzazioni. Per impostazione predefinita, le ACL sono disabilitate per nuovi bucket. È anche possibile disabilitare le ACL per i bucket esistenti. Se hai un bucket esistente che contiene già oggetti, dopo aver disabilitato le ACL, l'oggetto e le ACL bucket non fanno più parte del processo di valutazione dell'accesso. L'accesso è invece concesso o negato in base alle policy.

Prima di disabilitare le ACL, assicurati di eseguire la seguente procedura:

- Esamina la policy del bucket per assicurarti che copra tutti i modi in cui intendi concedere l'accesso al bucket al di fuori del tuo account.
- Ripristina le impostazioni di default del bucket ACL (controllo completo per il proprietario del bucket).

Dopo aver disabilitato le ACL, si verificano i seguenti comportamenti:

- Il bucket accetta solo richieste PUT che non specificano un ACL o richieste PUT con ACL di controllo completo del proprietario del bucket. Queste ACL includono l'ACL predefinita `bucket-owner-full-control` o forme equivalenti di questa ACL espresse in XML.
- Le applicazioni esistenti che supportano le ACL di controllo completo del proprietario del bucket non subiranno alcun impatto.
- PUTle richieste che contengono altri ACL (ad esempio, concessioni personalizzate a determinate Account AWS) hanno esito negativo e restituiscono un codice di stato HTTP `400 (Bad Request)` con il codice di errore. `AccessControlListNotSupported`

Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

Verifica che i bucket Amazon S3 utilizzino le policy corrette e non siano accessibili pubblicamente

A meno che non venga richiesto in maniera esplicita che gli utenti su Internet siano in grado di leggere o scrivere nel bucket S3, assicurati che il bucket S3 non sia pubblico. Di seguito sono riportate alcune delle fasi che è possibile eseguire per bloccare l'accesso pubblico:

- Utilizza Blocco dell'accesso pubblico S3. Con il Blocco dell'accesso pubblico, è possibile configurare facilmente controlli centralizzati per limitare l'accesso pubblico alle risorse Amazon S3. Questi controlli centralizzati vengono applicati a prescindere da come vengono create le risorse. Per ulteriori informazioni, consulta Blocco dell'accesso pubblico allo storage Amazon S3.
- Identifica le policy di bucket Amazon S3 che consentono l'uso di un'identità jolly, ad esempio `"Principal": "*"` (che significa di fatto "tutti"). Inoltre, cerca policy che consentono

un'azione jolly `"*"` (che di fatto consente all'utente di eseguire qualsiasi azione nel bucket Amazon S3).

- Allo stesso modo, cerca le liste di controllo degli accessi (ACL) dei bucket di Amazon S3 che forniscono lettura, scrittura o accesso completo a «Everyone» o «Qualsiasi utente autenticato». AWS
- Utilizza l'operazione API `ListBuckets` per eseguire la scansione di tutti i bucket Amazon S3. Quindi, utilizza `GetBucketAcl`, `GetBucketWebsite` e `GetBucketPolicy` per determinare se ciascun bucket dispone di controlli sugli accessi conformi e configurazione conforme.
- Utilizza AWS Trusted Advisor per ispezionare l'implementazione di Amazon S3.
- Valuta se implementare controlli di rilevamento continui utilizzando s3-bucket-public-read-prohibited e s3-bucket-public-write-prohibited gestito Regole di AWS Config.

Per ulteriori informazioni, consulta Identity and Access Management per Amazon S3.

Identifica potenziali minacce ai tuoi bucket Amazon S3 utilizzando Amazon GuardDuty

Amazon GuardDuty è un servizio di rilevamento delle minacce che identifica potenziali minacce per i tuoi account, contenitori, carichi di lavoro e dati all'interno del tuo AWS ambiente. Utilizzando modelli di machine learning (ML) e funzionalità di rilevamento di anomalie e minacce, Amazon monitora GuardDuty continuamente diverse fonti di dati per identificare e dare priorità ai potenziali rischi per la sicurezza e alle attività dannose nel tuo ambiente. Se abilitata GuardDuty, offre il rilevamento delle minacce per le fonti di dati fondamentali che includono eventi di AWS CloudTrail gestione, log di flusso VPC e log DNS. Per estendere il rilevamento delle minacce agli eventi del piano dati nei bucket S3, puoi abilitare la funzionalità S3 Protection. GuardDuty Questa funzionalità rileva minacce come l'esfiltrazione di dati e l'accesso sospetto ai bucket S3 tramite i nodi Tor. GuardDutystabilisce inoltre un normale modello di base nell'ambiente e, quando identifica un comportamento potenzialmente anomalo, fornisce informazioni contestuali per aiutarti a correggere il bucket o le credenziali S3 potenzialmente compromessi. AWS Per ulteriori informazioni, consulta. GuardDuty

Applica l'accesso con privilegi minimi

Quando concedi le autorizzazioni, puoi decidere quali autorizzazioni assegnare, a chi e per quali risorse Amazon S3. Puoi abilitare operazioni specifiche che desideri consentire su tali risorse. Pertanto, è consigliabile concedere solo le autorizzazioni necessarie richieste per eseguire un'attività. L'implementazione dell'accesso con privilegi minimi è fondamentale per ridurre i rischi di sicurezza e l'impatto risultante da errori o intenzioni dannose.

Gli strumenti seguenti sono disponibili per implementare l'accesso con privilegi minimi:

- Azioni politiche per Amazon S3 e Limiti delle autorizzazioni per le entità IAM
- Come funziona Amazon S3 con IAM
- Panoramica delle liste di controllo accessi (ACL)
- Policy di controllo dei servizi

Per indicazioni sugli aspetti da tenere in considerazione quando scegli uno o più dei meccanismi precedenti, consulta Identity and Access Management per Amazon S3.

Usa i ruoli IAM per applicazioni Servizi AWS che richiedono l'accesso ad Amazon S3

Affinché le applicazioni in esecuzione su Amazon EC2 o altro possano accedere Servizi AWS alle risorse Amazon S3, devono includere credenziali AWS valide nelle loro richieste API. AWS Consigliamo di non archiviare AWS le credenziali direttamente nell'applicazione o nell'istanza Amazon EC2. Si tratta di credenziali a lungo termine che non vengono automaticamente ruotate e potrebbero avere un impatto aziendale significativo se vengono compromesse.

Utilizza invece un ruolo IAM per gestire credenziali temporanee per le applicazioni o i servizi che devono accedere ad Amazon S3. Quando utilizzi un ruolo, non devi distribuire credenziali a lungo termine (come nome utente e password o chiavi di accesso) a un'istanza Amazon EC2 Servizio AWS o, ad esempio. AWS Lambda Il ruolo fornisce autorizzazioni temporanee che le applicazioni possono utilizzare quando effettuano chiamate ad altre risorse. AWS

Per ulteriori informazioni, consulta gli argomenti seguenti nella Guida per l'utente IAM:

- Ruoli IAM
- Scenari comuni per ruoli: utenti, applicazioni e servizi

Prendi in considerazione la crittografia dei dati inattivi

Per la protezione dei dati inattivi in Amazon S3 sono disponibili le opzioni seguenti:

- Crittografia lato server: tutti i bucket Amazon S3 hanno la crittografia configurata di default e tutti i nuovi oggetti caricati in un bucket S3 vengono crittografati automaticamente quando sono inattivi. La crittografia lato server con le chiavi gestite da Amazon S3 (SSE-S3) è la configurazione predefinita della crittografia per ogni bucket di Amazon S3. Per utilizzare un diverso tipo di crittografia, puoi specificare il tipo di crittografia lato server da utilizzare nelle richieste PUT S3 oppure impostare la configurazione di crittografia predefinita nel bucket di destinazione.

  Amazon S3 offre anche le seguenti opzioni di crittografia lato server:

  - Crittografia lato server con AWS Key Management Service () chiavi (SSE-KMS)AWS KMS

- Crittografia lato server a doppio livello con () chiavi (DSSE-KMS) AWS Key Management Service AWS KMS
- Crittografia lato server con chiavi fornite dal cliente (SSE-C)

Per ulteriori informazioni, consulta Protezione dei dati con la crittografia lato server.

- Crittografia lato client: esegui la crittografia dei dati dal lato client e carica i dati crittografati in Amazon S3. In questo caso, è l'utente a gestire la procedura di crittografia, nonché le chiavi e gli strumenti correlati. Come per la crittografia lato server, la crittografia lato client riduce i rischi crittografando i dati con una chiave che viene archiviata in un meccanismo diverso rispetto a quello utilizzato per archiviare i dati stessi.

  Amazon S3 fornisce più opzioni di crittografia lato client. Per ulteriori informazioni, consulta Protezione dei dati con la crittografia lato client.

Applica la crittografia dei dati in transito

È possibile utilizzare HTTPS (TLS) per impedire a potenziali aggressori di intercettare o manipolare il traffico di rete utilizzando o meno attacchi simili. person-in-the-middle Si consiglia di consentire solo connessioni crittografate su HTTPS (TLS) utilizzando la condizione aws:SecureTransport nelle policy di bucket Amazon S3.

> **Important**
>
> Consigliamo alla tua applicazione di non bloccare i certificati TLS di Amazon S3 poiché AWS non supporta il blocco di certificati pubblicamente attendibili. S3 rinnova automaticamente i certificati e il rinnovo può avvenire in qualsiasi momento prima della scadenza del certificato. Il rinnovo di un certificato genera una nuova coppia di chiavi pubblica-privata. Se hai aggiunto un certificato S3 che è stato recentemente rinnovato con una nuova chiave pubblica, non potrai connetterti a S3 finché l'applicazione non utilizzerà il nuovo certificato.

Inoltre, valuta se implementare controlli di rilevamento continui utilizzando la regola s3-bucket-ssl-requests-only gestita da AWS Config .

Valutazione dell'utilizzo di S3 Object Lock

Con S3 Object Lock, puoi archiviare gli oggetti utilizzando il modello "Write Once Read Many" (WORM). Il blocco oggetti S3 può contribuire a evitare l'eliminazione accidentale o

inappropriata dei dati. Ad esempio, puoi usare S3 Object Lock per proteggere i tuoi log. AWS CloudTrail

Per ulteriori informazioni, consulta Utilizzo del blocco oggetti S3.

Abilitazione del controllo delle versioni S3

Il controllo delle versioni S3 è un modo per conservare più versioni di un oggetto nello stesso bucket. Si può utilizzare questa funzione per conservare, recuperare e ripristinare qualsiasi versione di ogni oggetto archiviato nel bucket . Con la funzione Controllo delle versioni si può facilmente eseguire il ripristino dopo errori dell'applicazione e operazioni non intenzionali dell'utente.

Inoltre, valuta se implementare controlli di rilevamento continui utilizzando la regola s3-bucket-versioning-enabled gestita da AWS Config .

Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Valutazione dell'utilizzo della replica tra regioni S3

Sebbene Amazon S3 per impostazione predefinita archivi i dati in più zone di disponibilità geograficamente distanti, per soddisfare i requisiti di conformità potrebbe essere necessario archiviarli a distanze ancora maggiori. Con S3 Cross-Region Replication (CRR), puoi replicare i dati tra distanti per soddisfare questi requisiti. Regioni AWS CRR consente la copia automatica e asincrona di oggetti tra bucket diversi. Regioni AWS Per ulteriori informazioni, consulta Panoramica sulla replica degli oggetti.

> **Note**
>
> CRR richiede che il controllo delle versioni sia abilitato per i bucket S3 di origine e destinazione.

Inoltre, valuta se implementare controlli di rilevamento continui utilizzando la regola s3-bucket-replication-enabled gestita da AWS Config .

Valutazione dell'utilizzo degli endpoint VPC per l'accesso ad Amazon S3

Un endpoint Virtual Private Cloud (VPC) Amazon S3 è un'entità logica all'interno di un VPC che consente la connettività solo ad Amazon S3 Gli endpoint VPC impediscono al traffico di attraversare la rete Internet aperta.

Gli endpoint VPC per Amazon S3 offrono diversi modi per controllare l'accesso ai dati di Amazon S3:

- È possibile controllare le richieste, gli utenti o i gruppi autorizzati tramite un endpoint VPC specifico utilizzando policy del bucket S3.
- È possibile controllare quali VPC o endpoint VPC hanno accesso ai bucket S3 utilizzando le policy di bucket S3.
- Puoi impedire l'esfiltrazione di dati utilizzando un VPC che non dispone di un Internet gateway.

Per ulteriori informazioni, consulta Controllo dell'accesso dagli endpoint VPC con policy di bucket.

Utilizza servizi di sicurezza gestiti AWS per monitorare la sicurezza dei dati

Diversi servizi AWS di sicurezza gestiti possono aiutarti a identificare, valutare e monitorare i rischi di sicurezza e conformità per i tuoi dati Amazon S3. Questi servizi consentono anche di proteggere i dati da tali rischi. Questi servizi includono funzionalità di rilevamento, monitoraggio e protezione automatizzate progettate per scalare dalle risorse di Amazon S3 per una singola unità Account AWS a risorse per organizzazioni con migliaia di account.

Per ulteriori informazioni, consulta Monitoraggio della sicurezza dei dati con servizi AWS di sicurezza gestiti.

Best practice di monitoraggio e audit di Amazon S3

Le best practice seguenti per Amazon S3 consentono di rilevare potenziali debolezze e incidenti di sicurezza.

Identificazione e audit di tutti i bucket Amazon S3

L'identificazione degli asset IT è un aspetto essenziale di governance e sicurezza. È richiesta la visibilità di tutte le risorse Amazon S3 per valutare il loro assetto di sicurezza e intervenire su aree di debolezza potenziali. Per eseguire l'audit delle risorse, procedi come segue:

- Utilizza Tag Editor per identificare e applicare tag a risorse sensibili alla sicurezza e risorse sensibili al controllo; quindi, utilizza questi tag quando devi cercare le risorse. Per ulteriori informazioni, consulta Searching for Resources to Tag nella Tagging AWS Resources User Guide.
- Utilizza S3 Inventory per eseguire l'audit e creare report sullo stato di replica e crittografia degli oggetti per esigenze aziendali, di conformità e normative. Per ulteriori informazioni, consulta Amazon S3 Inventory.

- Crea gruppi di risorse per le risorse Amazon S3. Per ulteriori informazioni, consulta Che cosa sono i gruppi di risorse? nella Guida per l'utente di AWS Resource Groups .

Implementa il monitoraggio utilizzando strumenti AWS di monitoraggio

Il monitoraggio è una parte importante per mantenere l'affidabilità, la sicurezza, la disponibilità e le prestazioni di Amazon S3 e delle tue AWS soluzioni. AWS fornisce diversi strumenti e servizi per aiutarti a monitorare Amazon S3 e altri. Servizi AWS Ad esempio, puoi monitorare i CloudWatch parametri di Amazon per Amazon S3, in particolare `PutRequests` i parametri`GetRequests`,`4xxErrors`, `DeleteRequests` e. Per ulteriori informazioni, consulta Monitoraggio delle metriche con Amazon CloudWatch e Monitoraggio di Amazon S3.

Per un secondo esempio, consulta Esempio: attività del bucket Amazon S3. Questo esempio descrive come creare un CloudWatch allarme che viene attivato quando viene effettuata una chiamata API Amazon S3 o una policy del bucket, un ciclo di vita del bucket `DELETE` o una configurazione di replica del bucket o `PUT` verso un bucket ACL. `PUT`

Abilita la registrazione degli accessi al server Amazon S3

La registrazione degli accessi al server fornisce record dettagliati delle richieste che sono effettuate a un bucket. I log di accesso al server possono essere utili durante gli audit di sicurezza e accesso, per conoscere la base clienti e comprendere la fattura Amazon S3. Per istruzioni sull'abilitazione della registrazione degli accessi al server, consulta Registrazione delle richieste con registrazione dell'accesso al server.

Prendi in considerazione anche l'implementazione di controlli investigativi continui utilizzando la regola gestita. s3-bucket-logging-enabled AWS Config

Utilizza AWS CloudTrail

AWS CloudTrail fornisce un registro delle azioni intraprese da un utente, da un ruolo o da un utente Servizio AWS in Amazon S3. Puoi utilizzare le informazioni raccolte da CloudTrail per determinare quanto segue:

- La richiesta effettuata ad Amazon S3
- L'indirizzo IP dal quale è stata effettuata la richiesta
- L'utente che ha effettuato la richiesta
- L'ora in cui è stata effettuata la richiesta
- Dettagli aggiuntivi relativi alla richiesta

Ad esempio, è possibile identificare le CloudTrail voci relative alle PUT azioni che influiscono sull'accesso ai dati`PutBucketAcl`, in particolare`PutObjectAcl`,`PutBucketPolicy`, e`PutBucketWebsite`.

Quando si configura il Account AWS, CloudTrail è abilitato per impostazione predefinita. Puoi visualizzare gli eventi recenti nella CloudTrail console. Per creare un record continuo di attività ed eventi per i tuoi bucket Amazon S3, puoi creare un percorso nella console. CloudTrail Per ulteriori informazioni, consultare Registrazione di eventi di dati nella Guida per l'utente di AWS CloudTrail .

Quando crei un percorso, puoi configurare la registrazione degli eventi relativi CloudTrail ai dati. Gli eventi di dati sono le registrazioni delle operazioni eseguite per una risorsa o al suo interno. In Amazon S3, gli eventi relativi ai dati registrano l'attività delle API a livello di oggetto per singoli bucket. CloudTrail supporta un sottoinsieme di operazioni API a livello di oggetto Amazon S3, ad esempio`GetObject`, e. `DeleteObject` `PutObject` Per ulteriori informazioni su come CloudTrail funziona con Amazon S3, consulta. Registrazione delle chiamate API Amazon S3 tramite AWS CloudTrail Nella console di Amazon S3, puoi configurare i tuoi bucket S3 anche su Abilitazione della registrazione CloudTrail degli eventi per bucket e oggetti S3.

AWS Config fornisce una regola gestita (`cloudtrail-s3-dataevents-enabled`) che puoi utilizzare per confermare che almeno un CloudTrail trail registri gli eventi relativi ai dati per i tuoi bucket S3. Per ulteriori informazioni, consulta la sezione cloudtrail-s3-dataevents-enabled nella Guida per gli sviluppatori di AWS Config .

Attiva AWS Config

Diverse delle best practice elencate in questo argomento suggeriscono la creazione di regole. AWS Config AWS Config ti aiuta a valutare, controllare e valutare le configurazioni delle tue AWS risorse. AWS Config monitora le configurazioni delle risorse in modo da poter valutare le configurazioni registrate rispetto alle configurazioni sicure desiderate. Con AWS Config, è possibile effettuare le seguenti operazioni:

- Rivedere le modifiche nelle configurazioni e nelle relazioni tra le risorse AWS
- Investigare le cronologie dettagliate della configurazione delle risorse
- Determinare la conformità complessiva rispetto alle configurazioni specificate nelle linee guida interne

Using AWS Config può aiutarvi a semplificare il controllo della conformità, l'analisi della sicurezza, la gestione delle modifiche e la risoluzione dei problemi operativi. Per ulteriori informazioni,

consulta Configurazione AWS Config con la console nella Guida per gli AWS Config sviluppatori. Durante la specifica dei tipi di risorse da registrare, assicurati di includere le risorse Amazon S3.

> **Important**
>
> AWS Config managed rules supporta solo bucket generici durante la valutazione delle risorse Amazon S3. AWS Config non registra le modifiche alla configurazione per i bucket di directory. Per ulteriori informazioni, consulta AWS Config Managed Rules e List of AWS Config Managed Rules nella AWS Config Developer Guide.

Per un esempio di utilizzo AWS Config, consulta How to Use AWS Config to Monitor for and Respond to Amazon S3 Bucket Allowing Public Access sul blog sulla AWS sicurezza.

Scoprire dati sensibili utilizzando Amazon Macie

Amazon Macie è un servizio di sicurezza che rileva dati sensibili utilizzando il machine learning e la corrispondenza del modello. Macie fornisce visibilità sui rischi legati alla sicurezza dei dati e consente una protezione automatizzata da tali rischi. Con Macie, puoi automatizzare l'individuazione e la creazione di report dei dati sensibili nel tuo patrimonio di dati Amazon S3 per una migliore comprensione dei dati archiviati dall'organizzazione in S3.

Per individuare dati sensibili con Macie, puoi utilizzare criteri e tecniche integrati progettati per rilevare un elenco ampio e in continua espansione di tipi di dati sensibili per molti Paesi e regioni. Questi tipi di dati sensibili includono diversi tipi di informazioni di identificazione personale (PII), dati finanziari e dati delle credenziali. Puoi anche utilizzare criteri personalizzati: espressioni regolari che definiscono modelli di testo da abbinare e, facoltativamente, sequenze di caratteri e regole di prossimità per perfezionare i risultati.

Se Macie rileva dati sensibili in un oggetto S3, genera un risultato relativo alla sicurezza per informare l'utente. Questo risultato fornisce informazioni sull'oggetto interessato, i tipi e il numero di occorrenze dei dati sensibili individuati da Macie e dettagli aggiuntivi per facilitare l'analisi del bucket S3 e dell'oggetto interessati. Per ulteriori informazioni, consultare la Guida per l'utente di Amazon Macie.

Utilizzo di S3 Storage Lens

S3 Storage Lens è una funzionalità di analisi dell'archiviazione su cloud che puoi utilizzare per avere una panoramica completa a livello di organizzazione sull'utilizzo e sulle attività relative all'archiviazione di oggetti. S3 Storage Lens analizza i parametri di archiviazione per fornire

raccomandazioni contestuali che puoi usare per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati.

Con S3 Storage Lens puoi usare i parametri per generare approfondimenti, ad esempio per scoprire la quantità di spazio di archiviazione disponibile nell'intera organizzazione o quali sono i bucket e i prefissi caratterizzati da una crescita più rapida. Puoi utilizzare i parametri di Amazon S3 Storage Lens anche per individuare le opportunità di ottimizzazione dei costi, implementare le best practice di protezione dei dati e gestione degli accessi e migliorare le prestazioni dei carichi di lavoro delle applicazioni.

Ad esempio, puoi identificare i bucket che non hanno regole del ciclo di vita S3 per interrompere i caricamenti in più parti incompleti che risalgono a più di 7 giorni. Puoi anche individuare i bucket non conformi alle best practice di protezione dei dati, come quelli che usano la replica S3 o il controllo delle versioni S3. Per maggiori informazioni, consulta Informazioni su Amazon S3 Storage Lens.

Monitora i suggerimenti di sicurezza di AWS

È opportuno controllare regolarmente i consigli di sicurezza pubblicati in Trusted Advisor per il tuo Account AWS. In particolare, cerca gli avvisi relativi ai bucket Amazon S3 con "autorizzazioni di accesso aperte". Puoi eseguire questa operazione a livello di codice o utilizzando describe-trusted-advisor-checks.

Inoltre, monitora attivamente l'indirizzo e-mail principale registrato su ciascuno dei tuoi. Account AWS AWS utilizza questo indirizzo email per contattarti in merito a problemi di sicurezza emergenti che potrebbero interessarti.

AWS i problemi operativi di ampio impatto sono pubblicati sulla pagina AWS Health Dashboard - Stato del servizio. I problemi operativi sono anche pubblicati sui singoli account tramite AWS Health Dashboard. Per ulteriori informazioni, consulta la documentazione relativa ad AWS Health.

Monitoraggio della sicurezza dei dati con servizi AWS di sicurezza gestiti

Diversi servizi AWS di sicurezza gestiti possono aiutarti a identificare, valutare e monitorare i rischi di sicurezza e conformità per i tuoi dati Amazon S3. Consentono anche di proteggere i dati da tali rischi. Questi servizi includono funzionalità di rilevamento, monitoraggio e protezione automatizzate progettate per scalare dalle risorse di Amazon S3 per una singola unità Account AWS a risorse per organizzazioni che comprendono migliaia di utenti. Account AWS

AWS i servizi di rilevamento e risposta possono aiutarti a identificare potenziali configurazioni errate di sicurezza, minacce o comportamenti imprevisti, in modo da poter rispondere rapidamente ad attività potenzialmente non autorizzate o dannose nel tuo ambiente. AWS i servizi di protezione dei dati possono aiutarti a monitorare e proteggere dati, account e carichi di lavoro da accessi non autorizzati. Inoltre, consentono di individuare dati sensibili, come informazioni di identificazione personale (PII), nel tuo patrimonio di dati Amazon S3.

Per semplificare l'identificazione e la valutazione dei rischi di sicurezza e conformità dei dati, i servizi di sicurezza AWS gestiti generano risultati per segnalare potenziali eventi o problemi di sicurezza con i dati Amazon S3. I risultati forniscono dettagli rilevanti che possono essere utilizzati per analizzare, valutare e agire su questi rischi in base ai flussi di lavoro e alle policy di risposta agli eventi imprevisti. È possibile accedere direttamente ai dati dei risultati utilizzando ciascun servizio. Inoltre, è possibile inviare i dati ad altre applicazioni, servizi e sistemi, ad esempio il sistema SIEM (Security Incident and Event Management).

Per monitorare la sicurezza dei tuoi dati Amazon S3, prendi in considerazione l'utilizzo di questi servizi di AWS sicurezza gestiti.

Amazon GuardDuty

Amazon GuardDuty è un servizio di rilevamento delle minacce che monitora continuamente i tuoi carichi di lavoro Account AWS e quelli di lavoro alla ricerca di attività dannose e fornisce risultati di sicurezza dettagliati per visibilità e correzione.

Con la funzionalità di protezione S3 attiva GuardDuty, puoi configurare l'analisi degli eventi GuardDuty di AWS CloudTrail gestione e dei dati per le tue risorse Amazon S3. GuardDuty monitora quindi tali eventi alla ricerca di attività dannose e sospette. Per supportare l'analisi e identificare i potenziali rischi per la sicurezza, GuardDuty utilizza feed di intelligence sulle minacce e apprendimento automatico.

GuardDuty può monitorare diversi tipi di attività per le tue risorse Amazon S3. Ad esempio, gli eventi di CloudTrail gestione per Amazon S3 includono operazioni a livello di bucket, come`ListBuckets`, e. `DeleteBucket` `PutBucketReplication` CloudTrail gli eventi di dati per Amazon S3 includono operazioni a livello di oggetto, ad esempio, e`GetObject`. `ListObjects` `PutObject` Se GuardDuty rileva attività anomale o potenzialmente dannose, genera un risultato da inviare all'utente.

Per ulteriori informazioni, consulta Amazon S3 Protection in Amazon GuardDuty nella Amazon GuardDuty User Guide.

Amazon Detective

Amazon Detective semplifica il processo di analisi e consente di condurre indagini sulla sicurezza più rapide ed efficaci. Detective fornisce aggregazioni di dati, riepiloghi e contesto predefiniti che facilitano l'analisi e la valutazione della natura e dell'estensione dei possibili problemi di sicurezza.

Detective estrae automaticamente gli eventi basati sul tempo, come le chiamate API e i log di flusso di AWS CloudTrail Amazon VPC, per le tue risorse. AWS Inoltre, acquisisce i risultati generati da Amazon GuardDuty. Detective utilizza quindi machine learning, l'analisi statistica e la teoria dei grafi per generare visualizzazioni che consentono di condurre indagini sulla sicurezza efficaci più rapidamente.

Queste visualizzazioni forniscono una vista unificata, interattiva dei comportamenti delle risorse e delle interazioni tra di esse nel tempo. È possibile esplorare questo grafico del comportamento per esaminare possibili azioni dannose, come tentativi di accesso non riusciti o chiamate API sospette. È anche possibile vedere in che modo queste azioni interessano le risorse, come bucket e oggetti S3.

Per ulteriori informazioni, consultare la Guida di amministrazione di Amazon Detective.

Sistema di analisi degli accessi AWS IAM

AWS Identity and Access Management Access Analyzer (IAM Access Analyzer) può aiutarti a identificare le risorse condivise con un'entità esterna. Puoi anche utilizzare IAM Access Analyzer per convalidare le policy IAM in base alla grammatica e alle best practice delle policy e generare policy IAM basate sull'attività di accesso nei tuoi log. AWS CloudTrail

IAM Access Analyzer utilizza il ragionamento basato sulla logica per analizzare le politiche relative alle risorse nel tuo ambiente, come le bucket policy. AWS Con IAM Access Analyzer for S3, vieni avvisato quando un bucket S3 viene configurato per consentire l'accesso a chiunque sia

connesso a Internet o altro, compresi gli account esterni all'organizzazione. Account AWS Ad esempio, IAM Access Analyzer per S3 potrebbe segnalare che un bucket dispone di accesso in lettura o scrittura fornito tramite una lista di controllo degli accessi (ACL) del bucket, una policy del bucket, una policy del punto di accesso multi-regione o una policy del punto di accesso. Per ogni bucket pubblico o condiviso, vengono visualizzati risultati che indicano l'origine e il livello di accesso pubblico o condiviso. Con questi risultati puoi eseguire azioni correttive immediate e precise per ripristinare l'accesso del bucket desiderato.

Per ulteriori informazioni, consulta Revisione dell'accesso al bucket tramite IAM Access Analyzer per S3.

Amazon Macie

Amazon Macie è un servizio di sicurezza dei dati che rileva dati sensibili utilizzando machine learning e la corrispondenza del modello, fornisce visibilità sui rischi legati alla sicurezza dei dati e consente una protezione automatizzata da tali rischi.

Con Macie, è possibile automatizzare l'individuazione e la creazione di report dei dati sensibili nei bucket S3 per una migliore comprensione dei dati archiviati dall'organizzazione in Amazon S3. Per rilevare dati sensibili, è possibile utilizzare criteri e tecniche predefiniti forniti da Macie, criteri personalizzati definiti dall'utente o una combinazione dei due. Se Macie rileva dati sensibili in un oggetto S3, genera un risultato per informare l'utente. Questo risultato fornisce informazioni sul bucket e l'oggetto interessati, i tipi e il numero di occorrenze dei dati sensibili individuati da Macie e dettagli aggiuntivi per facilitare l'indagine.

Macie fornisce anche statistiche e altri dati che offrono informazioni dettagliate sullo stato di sicurezza dei dati di Amazon S3, inoltre, valuta e monitora automaticamente i bucket S3 per la sicurezza e il controllo degli accessi. Se Macie rileva un possibile problema con la sicurezza o la privacy dei dati dell'utente, ad esempio un bucket che diventa accessibile pubblicamente, genera un risultato per eseguire la verifica e la correzione, in base alle esigenze.

Per ulteriori informazioni, consultare la Guida per l'utente di Amazon Macie.

AWS Security Hub

AWS Security Hub è un servizio di gestione del livello di sicurezza che esegue controlli basati sulle migliori pratiche di sicurezza, aggrega avvisi e risultati provenienti da più fonti in un unico formato e consente la correzione automatica.

Security Hub raccoglie e fornisce dati sui risultati di sicurezza da soluzioni di AWS Partner Network sicurezza integrate Servizi AWS, tra cui Amazon Detective, Amazon GuardDuty, IAM

Access Analyzer e Amazon Macie. Genera inoltre i propri risultati eseguendo controlli di sicurezza continui e automatizzati basati sulle AWS migliori pratiche e sugli standard di settore supportati.

Security Hub esegue quindi la correlazione e consolida i risultati sui provider per aiutarti a stabilire le priorità ed elaborare i risultati più significativi. Inoltre, fornisce supporto per azioni personalizzate, che possono essere utilizzate per richiamare risposte o azioni correttive per classi specifiche di risultati.

Con Security Hub, puoi valutare lo stato di sicurezza e conformità delle tue risorse Amazon S3 nell'ambito di un'analisi più ampia del livello di sicurezza della tua organizzazione in singole regioni Regioni AWS e in più regioni. Ciò include l'analisi delle tendenze di sicurezza e l'identificazione dei problemi di sicurezza con priorità massima. È anche possibile aggregare i risultati di più Regioni AWS e monitorare ed elaborare i dati dei risultati aggregati di una singola regione.

Per ulteriori informazioni, consultare la sezione relativa ai controlli Amazon Simple Storage Service nella Guida per l'utente di AWS Security Hub .

Gestione dello storage in Amazon S3

Dopo aver creato bucket e caricato oggetti in Amazon S3, puoi gestire lo storage degli oggetti utilizzando funzionalità come la funzione Controllo delle versioni, le classi di storage, il blocco oggetti, le Batch Operations, la replica, i tag e altro ancora. Nelle sezioni seguenti vengono fornite informazioni dettagliate sulle funzionalità e sulle funzionalità di gestione dello storage disponibili in Amazon S3.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Argomenti

Utilizzo della funzione Controllo delle versioni nei bucket S3

La funzione Controllo delle versioni in Amazon S3 è un modo per conservare più versioni di un oggetto nello stesso bucket. Puoi utilizzare la funzione Controllo delle versioni S3 per conservare, recuperare e ripristinare qualsiasi versione di ogni oggetto archiviato nei tuoi bucket. Con la funzione Controllo delle versioni si può facilmente eseguire il ripristino dopo errori dell'applicazione e operazioni non intenzionali dell'utente. Quando abiliti la funzione Controllo delle versioni del bucket, se Amazon S3 riceve più richieste di scrittura per lo stesso oggetto contemporaneamente, vengono archiviati tutti gli oggetti.

I bucket con la funzione Controllo delle versioni abilitata consentono di ripristinare oggetti che sono stati eliminati o sovrascritti accidentalmente. Ad esempio, se elimini un oggetto, Amazon S3 inserisce un contrassegno di eliminazione invece di rimuovere l'oggetto in modo permanente. Il contrassegno di eliminazione diventa la versione corrente dell'oggetto. La sovrascrittura di un oggetto genera una nuove versione dell'oggetto nel bucket. È sempre possibile ripristinare la versione precedente. Per ulteriori informazioni, consulta Eliminazione di versioni di oggetti da un bucket con funzione Controllo delle versioni abilitata.

Per impostazione predefinita, la funzione Controllo delle versioni S3 è disabilitato nei bucket ed è necessario abilitarlo esplicitamente. Per ulteriori informazioni, consulta Abilitazione della funzione Controllo delle versioni sui bucket.

> **Note**
>
> - L'API SOAP non supporta la funzione Controllo delle versioni S3. Il supporto di SOAP su HTTP non viene più utilizzato, ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non sono supportate per SOAP.
> - A ogni versione archiviata e trasferita di un oggetto si applicano le tariffe Amazon S3 normali. Ogni versione di un oggetto è l'intero oggetto e non semplicemente la sua versione precedente con qualche differenza. Per questo motivo, se sono presenti tre versioni di un oggetto memorizzate verranno addebitati tre oggetti.

Bucket senza versione, con funzione Controllo delle versioni e con funzione Controllo delle versioni sospesa

I bucket possono trovarsi in uno dei tre stati:

- Senza versione (impostazione predefinita)
- Funzione Controllo delle versioni attivata
- Funzione Controllo delle versioni sospesa

Puoi abilitare e sospendere la funzione Controllo delle versioni a livello di bucket. Dopo aver abilitato la funzione Controllo delle versioni del bucket, non è possibile riportare il bucket nello stato senza versione. Tuttavia puoi sospendere la funzione Controllo delle versioni su tali bucket.

La funzione Controllo delle versioni si applica a tutti (mai solo ad alcuni) oggetti del bucket. Quando si abilita il controllo delle versioni in un bucket, tutti i nuovi oggetti vengono sottoposti al controllo versioni e viene assegnato un ID versione univoco. Gli oggetti già presenti nel bucket al momento in cui è stato abilitato il controllo delle versioni verranno successivamente sempre sottoposti al controllo versioni e verrà loro assegnato un ID versione univoco quando vengono modificati da richieste future. Tieni presente quanto segue:

- Gli oggetti che sono stati archiviati nel bucket prima dell'impostazione dello stato della funzione Controllo delle versioni hanno un ID versione `null`. Quando si abilita la funzione Controllo delle versioni, gli oggetti esistenti nel bucket non si modificano. Ciò che cambia è il modo in cui Amazon S3 gestirà gli oggetti delle richieste future. Per ulteriori informazioni, consulta Utilizzo di oggetti in un bucket che supporta la funzione Controllo delle versioni.
- Il proprietario del bucket (o un qualsiasi utente con le autorizzazioni appropriate) può sospendere la funzione Controllo delle versioni per interrompere l'accumulo di versioni. Quando si sospende la funzione Controllo delle versioni, gli oggetti esistenti nel bucket non si modificano. Ciò che cambia è il modo in cui Amazon S3 gestirà gli oggetti delle richieste future. Per ulteriori informazioni, consulta Utilizzo di oggetti di un bucket con funzione Controllo delle versioni sospesa.

Utilizzo della funzione Controllo delle versioni S3 con il ciclo di vita di S3

La funzione Controllo delle versioni degli oggetti consente, insieme al ciclo di vita di S3, di personalizzare il metodo di conservazione dei dati e di controllare i costi di storage. Per ulteriori informazioni, consulta Gestione del ciclo di vita dello storage. Per informazioni sulla creazione di configurazioni S3 Lifecycle utilizzando AWS Management Console,, AWS CLI AWS SDK o l'API REST, consulta. Impostazione di una configurazione del ciclo di vita su un bucket

> **Important**
>
> Se nel bucket senza funzione Controllo delle versioni è presente una configurazione del ciclo di vita per la scadenza dell'oggetto e si vuole mantenere lo stesso comportamento di eliminazione permanente che si applica quando la funzione Controllo delle versioni è abilitata, è necessario aggiungere una configurazione di scadenza non corrente. La configurazione del ciclo di vita per la scadenza non corrente gestisce le cancellazioni delle versioni non correnti dell'oggetto nel bucket abilitato per il controllo delle versioni. (Un bucket abilitato per le versioni mantiene una versione dell'oggetto corrente e zero o più versioni dell'oggetto non correnti.) Per ulteriori informazioni, consulta Impostazione di una configurazione del ciclo di vita su un bucket.

Per informazioni sull'utilizzo della funzione Controllo delle versioni S3, fai riferimento agli argomenti di seguito.

Argomenti

Come funzionano il Controllo delle versioni S3

Puoi utilizzare il controllo delle versioni S3 per mantenere più versioni di un oggetto in un unico bucket e ripristinare gli oggetti che vengono accidentalmente eliminati o sovrascritti. Ad esempio, se applichi il controllo delle versioni S3 a un bucket, si verificano le seguenti modifiche:

- Se anziché rimuovere un oggetto in modo permanente lo elimini, Amazon S3 inserisce un contrassegno di eliminazione che diventa la versione corrente dell'oggetto. È quindi possibile ripristinare la versione precedente. Per ulteriori informazioni, consulta Eliminazione di versioni di oggetti da un bucket con funzione Controllo delle versioni abilitata.
- Se sovrascrivi un oggetto, Amazon S3 aggiunge una nuova versione dell'oggetto nel bucket. La versione precedente rimane nel bucket e diventa una versione non corrente. Puoi ripristinare la versione precedente.

> **Note**
>
> A ogni versione archiviata e trasferita di un oggetto si applicano le tariffe Amazon S3 normali. Ogni versione di un oggetto è l'intero oggetto e non la sua versione precedente con qualche differenza. Per questo motivo, se sono presenti tre versioni di un oggetto memorizzate verranno addebitati tre oggetti.

A ogni bucket S3 creato è associata una sottorisorsa per la funzione Controllo delle versioni. (Per ulteriori informazioni, consulta Opzioni di configurazione dei bucket.) Per impostazione predefinita, il bucket è senza versione, di conseguenza la sottorisorsa per la funzione Controllo delle versioni archivia una configurazione vuota della funzione Controllo delle versioni.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</VersioningConfiguration>
```

Per abilitare il controllo delle versioni, puoi inviare una richiesta ad Amazon S3 con una configurazione del controllo delle versioni con lo stato `Enabled`.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Per sospendere la funzione Controllo delle versioni, si imposterà il valore dello stato su `Suspended`.

> **Note**
>
> Quando abiliti il controllo delle versioni in un bucket per la prima volta, potrebbe essere necessario un breve periodo di tempo per la propagazione completa della modifica. Ti consigliamo di attendere 15 minuti dopo aver abilitato il controllo delle versioni prima di eseguire operazioni di scrittura (`PUT` o `DELETE`) sugli oggetti nel bucket.

Il proprietario del bucket e tutti gli utenti autorizzati AWS Identity and Access Management (IAM) possono abilitare il controllo delle versioni. Il proprietario del bucket è colui Account AWS che ha creato il bucket. Per ulteriori informazioni sulle autorizzazioni, consultare Identity and Access Management per Amazon S3.

Per ulteriori informazioni sull'attivazione e la disabilitazione del controllo delle versioni di S3 utilizzando l'API AWS Management Console, AWS Command Line Interface (AWS CLI) o REST, consulta. the section called “Abilitazione della funzione Controllo delle versioni sui bucket”

Argomenti

ID versione

Se si abilita la funzione Controllo delle versioni del bucket, Amazon S3 genera automaticamente un ID versione univoco per l'oggetto archiviato. Ad esempio, un bucket può contenere due oggetti con la stessa chiave (nome dell'oggetto) ma ID versione diverso, ad esempio `photo.gif` (versione 111111) e `photo.gif` (versione 121212).
Diagramma che mostra un bucket abilitato al controllo delle versioni con due oggetti con la stessa chiave ma ID di versione diversi.

Ogni oggetto ha un ID versione, indipendentemente dal fatto che Controllo delle versioni S3 sia abilitato o meno. Se il controllo delle versioni S3 non è abilitato, Amazon S3 imposta il valore dell'ID versione su `null`. Se si attiva la funzione Controllo delle versioni S3, Amazon S3 assegna un valore ID versione per l'oggetto. Questo valore distingue l'oggetto dalle altre versioni della stessa chiave.

Quando si attiva la funzione Controllo delle versioni S3 in un bucket esistente, gli oggetti già archiviati nel bucket rimangono invariati. Gli ID versione (`null`), il contenuto e le autorizzazioni non sono cambiati. Dopo aver abilitato il controllo delle versioni S3, ogni oggetto aggiunto al bucket ottiene un ID versione che lo distingue dalle altre versioni della stessa chiave.

Solo Amazon S3 genera gli ID versione che non possono essere modificati. Gli ID di versione sono stringhe opache codificate in Unicode, UTF-8, pronte per l'URL, non lunghe più di 1.024 byte. Di seguito è riportato un esempio:

```
3sL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo
```

> **Note**
>
> Per semplicità, gli altri esempi in questo argomento utilizzano ID molto più brevi.

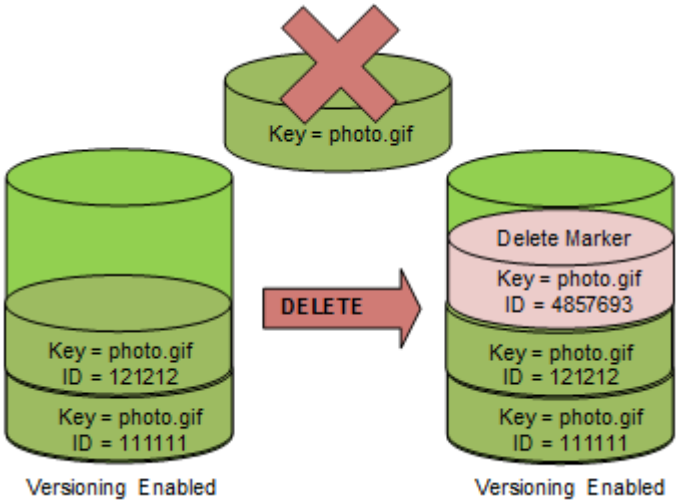
Flussi di lavoro per la funzione Controllo delle versioni

Se si esegue l'operazione PUT su un oggetto in un bucket abilitato per il controllo delle versioni, la versione non corrente non viene sovrascritta. Come mostrato nella seguente figura, quando viene eseguita un'operazione PUT per una nuova versione di `photo.gif` in un bucket che già contiene un oggetto con lo stesso nome:

- L'oggetto originale (ID = 111111) rimane nel bucket.
- Amazon S3 genera un nuovo ID versione (121212) e aggiunge nel bucket questa versione più recente dell'oggetto.

Con questa funzionalità è possibile recuperare la versione precedente di un oggetto che è stato accidentalmente sovrascritto o eliminato.

Quando esegui l'operazione `DELETE` per un oggetto, tutte le versioni rimangono nel bucket e Amazon S3 inserisce un contrassegno di eliminazione, come mostrato nell'illustrazione seguente.



Il contrassegno di eliminazione diventa la versione corrente dell'oggetto. Per default, le richieste `GET` recuperano la versione più recente archiviata. L'esecuzione di una richiesta `GET Object` quando la versione corrente è un contrassegno di eliminazione restituisce un errore `404 Not Found`, come mostrato nell'illustrazione seguente.

È possibile, tuttavia, eseguire un'operazione `GET` su una versione non corrente di un oggetto specificando l'ID versione corrispondente. Nell'illustrazione seguente viene eseguita un'operazione `GET` su una versione specifica dell'oggetto (111111). Amazon S3 restituisce la versione dell'oggetto anche se non è la versione corrente.

Per ulteriori informazioni, consulta Recupero delle versioni degli oggetti da un bucket con funzione Controllo delle versioni abilitata.

È possibile eliminare in modo permanente un oggetto specificando la versione che si desidera eliminare. Solo il proprietario di un bucket Amazon S3 o un utente IAM autorizzato può eliminare definitivamente una versione. Se l'operazione `DELETE` specifica `versionId`, la versione dell'oggetto viene eliminata definitivamente e Amazon S3 non inserisce un contrassegno di eliminazione.

Puoi aggiungere un ulteriore livello di sicurezza configurando un bucket per abilitare l'eliminazione con autenticazione a più fattori (MFA). Quando abiliti l'eliminazione MFA per un bucket, il proprietario del bucket deve includere due tipi di autenticazione in qualsiasi richiesta per eliminare una versione o modificare lo stato del controllo delle versioni del bucket. Per ulteriori informazioni, consulta Configurazione dell'eliminazione di MFA.

Quando vengono create nuove versioni per un oggetto?

Le nuove versioni vengono create solo quando esegui l'operazione `PUT` per un nuovo oggetto. Tieni presente che alcune azioni, come `CopyObject`, funzionano implementando un'operazione `PUT`.

Alcune azioni che modificano l'oggetto corrente non creano una nuova versione perché non eseguono l'operazione `PUT` di un nuovo oggetto. Ciò include azioni come la modifica dei tag su un oggetto.

> **Important**
>
> Se rilevi un aumento significativo nel numero di risposte HTTP 503 (servizio non disponibile) ricevute per le richieste `PUT` o `DELETE` di Amazon S3 in un bucket con il controllo delle versioni S3 abilitato, è possibile che esistano uno o più oggetti nel bucket per i quali sono presenti milioni di versioni. Per ulteriori informazioni, consulta la sezione sul controllo delle versioni S3 in Risoluzione dei problemi.

Abilitazione della funzione Controllo delle versioni sui bucket

È possibile utilizzare la funzione Controllo delle versioni S3 per mantenere più versioni di un oggetto in un bucket. Questa sezione fornisce esempi di come abilitare il controllo delle versioni su un bucket utilizzando la console, l'API REST, gli AWS SDK e (). AWS Command Line Interface AWS CLI

> **Note**
>
> Se abiliti il controllo delle versioni su un bucket per la prima volta, potrebbero essere necessari fino a 15 minuti prima che la modifica venga propagata completamente. Consigliamo di attendere 15 minuti dopo aver abilitato il controllo delle versioni prima di eseguire operazioni di scrittura (PUT o DELETE) sugli oggetti nel bucket. Le operazioni di scrittura eseguite prima del completamento di questa conversione possono essere applicate agli oggetti senza versione.

Per ulteriori informazioni sulla funzione Controllo delle versioni S3, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3. Per informazioni sull'utilizzo di oggetti che si trovano in bucket con la funzione Controllo delle versioni abilitata, consulta Utilizzo di oggetti in un bucket che supporta la funzione Controllo delle versioni.

Per ulteriori informazioni su come utilizzare la funzionalità S3 di controllo delle versioni per proteggere i dati, consulta Tutorial: Protezione dei dati su Amazon S3 da eliminazioni accidentali o bug delle applicazioni mediante le funzionalità S3 di controllo delle versioni, blocco degli oggetti e replica.

A ogni bucket S3 creato è associata una sottorisorsa per la funzione Controllo delle versioni. (Per ulteriori informazioni, consulta Opzioni di configurazione dei bucket.) Per impostazione predefinita, il bucket è senza versione, di conseguenza la sottorisorsa per la funzione Controllo delle versioni archivia una configurazione vuota della funzione Controllo delle versioni.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</VersioningConfiguration>
```

Per abilitare la funzione Controllo delle versioni, è possibile inviare una richiesta ad Amazon S3 con una configurazione della funzione che include lo stato.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Per sospendere la funzione Controllo delle versioni, si imposterà il valore dello stato su `Suspended`.

Il proprietario del bucket e tutti gli utenti autorizzati possono abilitare il controllo delle versioni. Il proprietario del bucket è colui Account AWS che ha creato il bucket (l'account root). Per ulteriori informazioni sulle autorizzazioni, consultare Identity and Access Management per Amazon S3.

Le sezioni seguenti forniscono maggiori dettagli sull'abilitazione del controllo delle versioni di S3 utilizzando la console e gli SDK AWS CLI. AWS

Utilizzo della console S3

Segui questi passaggi per utilizzare per abilitare il controllo delle versioni su AWS Management Console un bucket S3.

Per abilitare o disabilitare la funzione Controllo delle versioni in un bucket S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket name (Nome bucket) scegliere il nome del bucket per il quale si desidera abilitare la funzione Controllo delle versioni.
3. Scegliere Properties (Proprietà).
4. In Bucket Versioning (Funzione Controllo delle versioni del bucket) scegliere Edit (Modifica).
5. Scegliere Suspend (Sospendi) o Enable (Abilita), quindi scegliere Save changes (Salva modifiche).

> **Note**
>
> È possibile utilizzare l' AWS autenticazione a più fattori (MFA) con il controllo delle versioni. Quando utilizzi l'MFA con il controllo delle versioni, devi fornire le tue chiavi Account AWS di accesso e un codice valido dal dispositivo MFA dell'account per eliminare definitivamente una versione dell'oggetto o sospendere o riattivare il controllo delle versioni.
> Per utilizzare l'autenticazione MFA con la funzione Controllo delle versioni, abilita `MFA Delete`. Non è possibile abilitare `MFA Delete` utilizzando la AWS Management Console. È necessario utilizzare () o l'API AWS Command Line Interface .AWS CLI Per ulteriori informazioni, consulta Configurazione dell'eliminazione di MFA.

Utilizzando il AWS CLI

L'esempio seguente abilita la funzione Controllo delle versioni su un bucket S3.

```
aws s3api put-bucket-versioning --bucket example-s3-bucket1 --versioning-configuration
 Status=Enabled
```

L'esempio seguente abilita la funzione Controllo delle versioni S3 e l'eliminazione dell'autenticazione a più fattori (MFA) su un bucket.

```
aws s3api put-bucket-versioning --bucket example-s3-bucket1 --versioning-configuration
 Status=Enabled,MFADelete=Enabled --mfa "SERIAL 123456"
```

> **Note**
>
> L'utilizzo dell'eliminazione di MFA richiede un dispositivo di autenticazione fisico o virtuale approvato. Per ulteriori informazioni sull'utilizzo dell'eliminazione di MFA in Amazon S3, consulta Configurazione dell'eliminazione di MFA.

Per ulteriori informazioni sull'abilitazione del controllo delle versioni utilizzando il AWS CLI, vedere put-bucket-versioningnel AWS CLI Command Reference.

Utilizzo degli SDK AWS

Gli esempi seguenti abilitano il controllo delle versioni su un bucket e quindi recuperano lo stato del controllo delle versioni utilizzando and the. AWS SDK for Java AWS SDK for .NET Per informazioni sull'utilizzo di altri SDK AWS , consulta il Centro Developer di AWS.

.NET

Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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

Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
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

Nel seguente codice Python di esempio viene creato un bucket Amazon S3, viene abilitato per il controllo delle versioni e viene configurato un ciclo di vita che fa scadere le versioni degli oggetti non simultanee dopo 7 giorni.

```
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
```

Configurazione dell'eliminazione di MFA

Quando si utilizza la funzione Controllo delle versioni S3 nei bucket Amazon S3, puoi aggiungere un altro livello di sicurezza configurando un bucket per abilitare l'eliminazione MFA (autenticazione a più fattori). In tal caso, il proprietario del bucket deve includere due tipi di autenticazione in qualsiasi richiesta per eliminare una versione o modificare lo stato della funzione Controllo delle versioni del bucket.

La cancellazione MFA richiede autenticazione aggiuntiva per le seguenti operazioni:

- Modifica dello stato della funzione Controllo delle versioni del bucket
- Eliminazione permanente della versione di un oggetto

La cancellazione MFA richiede due forme di autenticazione contemporaneamente:

- Le credenziali di sicurezza
- la sequenza di un numero di serie valido, uno spazio e il codice a sei cifre visualizzato sul dispositivo di autenticazione approvato.

La cancellazione MFA fornisce così una protezione ulteriore, ad esempio se le credenziali di sicurezza fossero compromesse. L'eliminazione di MFA può aiutare a prevenire le eliminazioni accidentali dei bucket richiedendo all'utente che avvia l'azione di eliminazione di dimostrare il possesso fisico di un dispositivo MFA con un codice MFA e aggiungendo un ulteriore livello di interazione e sicurezza all'azione di eliminazione.

Per identificare i bucket con la funzionalità di eliminazione dell'autenticazione a più fattori (MFA) abilitata, puoi utilizzare i parametri di Amazon S3 Storage Lens. S3 Storage Lens è una funzionalità di analisi dell'archiviazione su cloud che puoi utilizzare per avere una panoramica completa a livello di organizzazione sull'utilizzo e sulle attività relative all'archiviazione di oggetti. Per ulteriori informazioni, consulta Valutazione dell'attività e dell'utilizzo dello storage con S3 Storage Lens. Per un elenco completo dei parametri, consulta Glossario dei parametri di S3 Storage.

Il proprietario del bucket, chi Account AWS ha creato il bucket (account root) e tutti gli utenti autorizzati possono abilitare il controllo delle versioni. Tuttavia, solo il proprietario del bucket (account root) può abilitare l'eliminazione di MFA. Per ulteriori informazioni, consulta la sezione Protezione dell'accesso all' AWS utilizzo della MFA nel blog AWS sulla sicurezza.

> **Note**
>
> Per utilizzare l'eliminazione MFA con la funzione Controllo delle versioni, abilita `MFA Delete`. Tuttavia, non è possibile abilitare `MFA Delete` l'utilizzo la AWS Management Console. È necessario utilizzare AWS Command Line Interface (AWS CLI) o l'API.
>
> Per esempi sull'utilizzo dell'eliminazione MFA con il controllo delle versioni, consulta la sezione degli esempi nell'argomento Abilitazione della funzione Controllo delle versioni sui bucket.
>
> Non puoi utilizzare l'eliminazione MFA con le configurazioni del ciclo di vita. Per ulteriori informazioni sulle configurazioni del ciclo di vita e sul modo in cui interagiscono con altre configurazioni, consulta Configurazioni del ciclo di vita e altre configurazioni del bucket.

Per abilitare o disabilitare la cancellazione MFA si ricorre alla stessa API utilizzata per configurare la funzione Controllo delle versioni di un bucket. Amazon S3 archivia la configurazione della cancellazione MFA nella stessa sottorisorsa della funzione Controllo delle versioni che contiene lo stato della funzione Controllo delle versioni del bucket.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>VersioningState</Status>
```

```
  <MfaDelete>MfaDeleteState</MfaDelete>
</VersioningConfiguration>
```

Per usare la cancellazione MFA si può utilizzare un dispositivo MFA fisico o virtuale per generare un codice di autenticazione. L'esempio seguente mostra il codice di autenticazione generato visualizzato su un dispositivo hardware.

La cancellazione MFA e l'accesso all'API protetto con autenticazione MFA sono caratteristiche destinate a offrire protezione in vari scenari. La cancellazione MFA viene configurata su un bucket per far sì che i dati del bucket non possano essere eliminati accidentalmente. Si utilizza l'accesso all'API protetto con autenticazione MFA per forzare un altro fattore di autenticazione (codice MFA) durante l'accesso a risorse Amazon S3 sensibili. Puoi richiedere che qualsiasi operazione su tali risorse di Amazon S3 venga eseguita fornendo credenziali temporanee create utilizzando MFA. Per un esempio, consulta Richiesta dell'autenticazione a più fattori (MFA).

Per ulteriori informazioni su come acquistare e attivare un dispositivo di autenticazione, consulta Autenticazione a più fattori.

Abilitazione del controllo delle versioni S3 e configurazione dell'eliminazione MFA

Utilizzando il AWS CLI

L'esempio seguente abilita la funzione Controllo delle versioni S3 e l'eliminazione dell'autenticazione a più fattori (MFA) su un bucket.

```
aws s3api put-bucket-versioning --bucket example-s3-bucket1 --versioning-configuration
 Status=Enabled,MFADelete=Enabled --mfa "SERIAL 123456"
```

Utilizzo di REST API

Per ulteriori informazioni su come specificare l'eliminazione MFA utilizzando l'API REST di Amazon S3, consulta PutBucketVersioningAmazon Simple Storage Service API Reference.

Utilizzo di oggetti in un bucket che supporta la funzione Controllo delle versioni

Gli oggetti che sono stati archiviati nel bucket Amazon S3 prima dell'impostazione dello stato della funzione Controllo delle versioni hanno un ID versione `null`. Quando si abilita la funzione Controllo delle versioni, gli oggetti esistenti nel bucket non si modificano. Ciò che cambia è il modo in cui Amazon S3 gestirà gli oggetti delle richieste future.

Trasferimento delle versioni di un oggetto

È possibile definire regole di configurazione del ciclo di vita per gli oggetti con un ciclo di vita ben definito per trasferire le versioni di tali oggetti alla classe di archiviazione S3 Glacier Flexible Retrieval (Recupero flessibile S3 Glacier) in uno specifico momento del ciclo di vita. Per ulteriori informazioni, consulta Gestione del ciclo di vita dello storage.

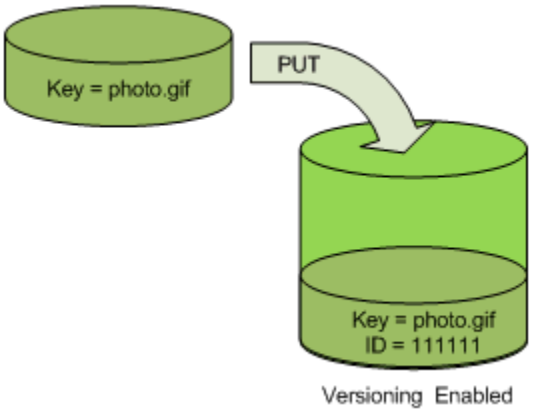
Gli argomenti di questa sezione illustrano varie operazioni sugli oggetti di un bucket che supporta la funzione Controllo delle versioni. Per ulteriori informazioni sulla funzione Controllo delle versioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Argomenti

Aggiunta di oggetti a bucket che supportano la funzione Controllo delle versioni

Dopo aver abilitato la funzione Controllo delle versioni del bucket, Amazon S3 aggiungerà automaticamente un ID versione univoco a ogni oggetto archiviato (utilizzando `PUT`, `POST` o `CopyObject`) nel bucket.

La figura seguente mostra l'aggiunta di un ID univoco a un oggetto da parte di Amazon S3 quando l'oggetto viene aggiunto a un bucket con la funzione Controllo delle versioni abilitata.



> **Note**
>
> I valori degli ID versione assegnati da Amazon S3 sono compatibili con l'URL (possono essere inclusi in un URI).

Per ulteriori informazioni sulla funzione Controllo delle versioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3. Puoi aggiungere versioni di oggetti a un bucket abilitato al controllo delle versioni utilizzando la console, gli SDK e l'API REST. AWS

Utilizzo della console

Per istruzioni, consulta Caricamento degli oggetti.

Uso dell'SDKs AWS

Per esempi di caricamento di oggetti utilizzando gli AWS SDK per Java, .NET e PHP, consulta. Caricamento degli oggetti Gli esempi di caricamento di oggetti in bucket senza versione e con funzione Controllo delle versioni abilitata sono identici ma, nel caso dei bucket con funzione Controllo delle versioni abilitata, Amazon S3 assegna un numero di versione. Negli altri casi il numero di versione è null.

Per informazioni sull'utilizzo di altri AWS SDK, consulta il Developer Center.AWS

Utilizzo di REST API

Per aggiungere oggetti a bucket che supportano la funzione Controllo delle versioni

1. Abilitare la funzione Controllo delle versioni del bucket tramite una richiesta `PutBucketVersioning`.

Per ulteriori informazioni, consulta PutBucketVersioning in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

2. Inviare una richiesta `PUT`, `POST` o `CopyObject` per memorizzare un oggetto nel bucket.

Quando si aggiunge un oggetto a un bucket con la funzione Controllo delle versioni abilitata, Amazon S3 restituisce l'ID versione dell'oggetto nell'intestazione di risposta `x-amz-version-id`, come mostrato nell'esempio di seguito.

```
x-amz-version-id: 3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY
```

Elenchi di oggetti in un bucket che supporta la funzione Controllo delle versioni

Questa sezione fornisce esempi di elenchi di versioni di oggetti di un bucket con funzione Controllo delle versioni abilitata. Amazon S3 archivia le informazioni sulla versione di un oggetto nella sottorisorsa versioni associata al bucket. Per ulteriori informazioni, consulta Opzioni di configurazione dei bucket. Per elencare gli oggetti in un bucket con il controllo delle versioni abilitato, è necessario disporre dell'autorizzazione `ListBucketVersions`.

Utilizzo della console S3

Segui questi passaggi per utilizzare la console di Amazon S3 per visualizzare le varie versioni di un oggetto.

Per visualizzare più versioni di un oggetto

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegliere il nome del bucket contenente l'oggetto.
3. Per visualizzare un elenco delle versioni degli oggetti nel bucket, scegli l'opzione Show versions (Mostra versioni).

   Per ogni versione dell'oggetto, la console mostra un ID versione univoco, la data e l'ora di creazione della versione e altre proprietà. Gli oggetti archiviati nel bucket prima dell'impostazione dello stato della funzione Controllo delle versioni hanno un ID versione null.

   Per elencare gli oggetti senza le versioni, scegliere l'opzione List versions (Elenca versioni) .

Puoi anche visualizzare, scaricare ed eliminare le versioni degli oggetti nel riquadro di panoramica sull'oggetto della console. Per ulteriori informazioni, consulta Visualizzazione della panoramica di un oggetto nella console di Amazon S3.

> **Note**
>
> Per accedere a versioni di oggetti precedenti a 300 versioni, è necessario utilizzare la AWS CLI o l'URL dell'oggetto.

> **Important**
>
> È possibile annullare l'eliminazione di un oggetto solo se è stato eliminato come ultima versione (corrente). Non è possibile annullare l'eliminazione della versione precedente di un oggetto eliminato. Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Utilizzo degli SDK AWS

Gli esempi di questa sezione mostrano come recuperare un elenco di oggetti da un bucket con funzione Controllo delle versioni abilitata. Ogni richiesta restituisce fino a 1.000 versioni, a meno che non sia stato specificato un valore inferiore. Se il bucket contiene un numero di versioni superiore a tale limite, sarà necessario inviare una serie di richieste per recuperare un elenco di tutte le versioni. Questo processo di restituzione di risultati in "pagine" è chiamato paginazione.

Per illustrare il funzionamento della paginazione, gli esempi limitano ogni risposta a due versioni di un oggetto. Dopo aver recuperato la prima pagina di risultati, ogni esempio verifica se l'elenco delle versioni è troncato. In caso affermativo, l'esempio continua recuperando pagine fino al recupero di tutte le versioni.

> **Note**
>
> Gli esempi seguenti operano anche con un bucket che non ha la funzione Controllo delle versioni abilitata o per gli oggetti che non hanno versioni specifiche. In questi casi Amazon S3 restituisce l'elenco di oggetti con la versione ID `null`.

Per informazioni sull'utilizzo di altri AWS SDK, consulta il AWS Developer Center.

Java

Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
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

Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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);
            }
        }
    }
}
```

Utilizzo di REST API

Example - Un elenco di tutte le versioni degli oggetti di un bucket

Per visualizzare un elenco di tutte le versioni degli oggetti in un bucket, utilizzare la sottorisorsa `versions` in una richiesta `GET Bucket`. Amazon S3 può recuperare fino a 1.000 oggetti e ogni versione di oggetto è conteggiata interamente come oggetto. Quindi se un bucket contiene due chiavi (ad esempio, `photo.gif` e `picture.jpg`) e la prima ha 990 versioni mentre la seconda ne ha 400, con una singola richiesta si potrebbero recuperare tutte le 990 versioni `photo.gif` e solo le 10 più recenti di `picture.jpg`.

Amazon S3 restituisce le versioni degli oggetti nell'ordine inverso rispetto a come sono state archiviate, ovvero l'ultima verrà restituita per prima.

Nella richiesta `GET Bucket`, includere la sottorisorsa `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 - Recupero di tutte le versioni di una chiave

Per recuperare un sottoinsieme di versioni di un oggetto, usa i parametri di richiesta per `GET Bucket`. Per ulteriori informazioni, consulta GET Bucket.

1. Impostare il parametro `prefix` sulla chiave dell'oggetto che si desidera recuperare.
2. Inviare una richiesta `GET Bucket` utilizzando la sottorisorsa `versions` e `prefix`.

```
GET /?versions&prefix=objectName HTTP/1.1
```

Example - Recupero di oggetti tramite un prefisso

Nell'esempio seguente vengono recuperati gli oggetti la cui chiave è o inizia con `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=
```

Si possono utilizzare altri parametri di richiesta per recuperare un sottoinsieme di tutte le versioni dell'oggetto. Per ulteriori informazioni, consulta GET Bucket in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Example - Recupero di un elenco di oggetti aggiuntivi se la risposta viene troncata

Se il numero di oggetti che possono essere restituiti in una richiesta `GET` supera il valore di `max-keys`, la risposta conterrà `<isTruncated>true</isTruncated>` e includerà la prima chiave (in `NextKeyMarker`) e il primo ID versione (in `NextVersionIdMarker`) che soddisfano la richiesta, ma che non sono stati restituiti. Si utilizzano i valori restituiti come posizione di inizio di una richiesta successiva per recuperare gli ulteriori oggetti che soddisfano la richiesta `GET`.

Utilizzare la procedura seguente per recuperare gli ulteriori oggetti di un bucket che soddisfano la richiesta `GET Bucket versions` originaria. Per ulteriori informazioni su `key-marker`, `version-id-marker`, `NextKeyMarker` e `NextVersionIdMarker`, consulta la sezione GET Bucket nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Di seguito sono riportate le risposte aggiuntive che soddisfano la richiesta `GET` originale:

- Impostare il valore di `key-marker` sulla chiave restituita in `NextKeyMarker` nella risposta precedente.

- Impostare il valore di `version-id-marker` sull'ID versione restituito in `NextVersionIdMarker` nella risposta precedente.
- Inviare una richiesta `GET Bucket versions` utilizzando `key-marker` e `version-id-marker`.

Example - Recupero di oggetto che iniziano con la chiave e l'ID versione specificati

```
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=
```

Utilizzo del AWS CLI

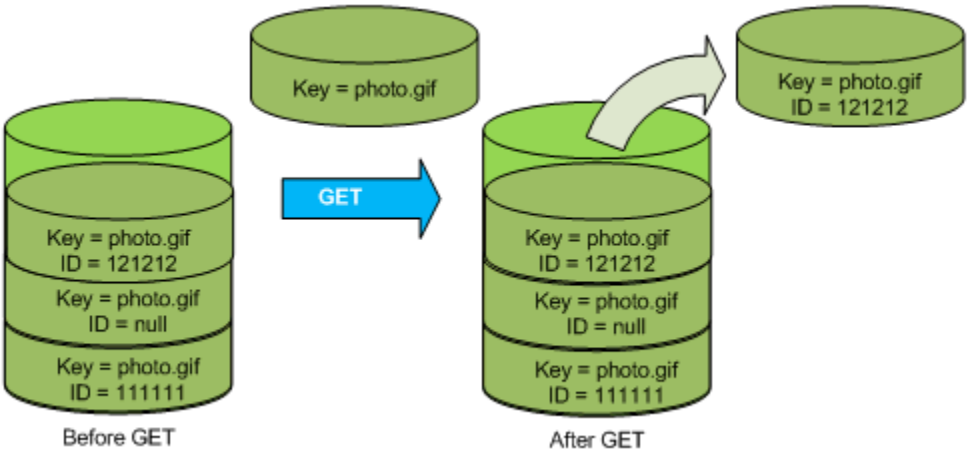
Il comando seguente restituisce i metadati relativi a tutte le versioni degli oggetti in un bucket.

```
aws s3api list-object-versions --bucket example-s3-bucket1
```

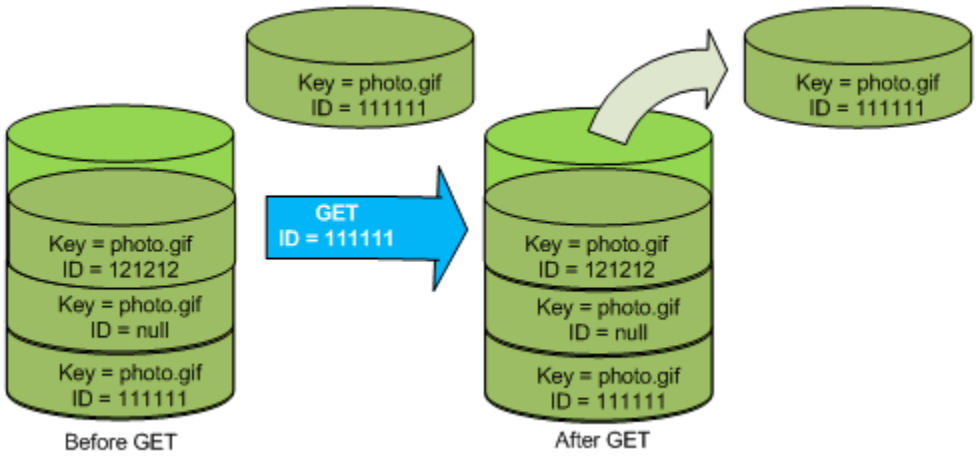
Per ulteriori informazioni su `list-object-versions`, consulta list-object-versions nel Riferimento ai comandi AWS CLI .

Recupero delle versioni degli oggetti da un bucket con funzione Controllo delle versioni abilitata

La funzione Controllo delle versioni in Amazon S3 è un modo per mantenere più varianti di un oggetto nello stesso bucket. Una richiesta `GET` semplice consente di recuperare la versione corrente di un oggetto. La figura seguente mostra come `GET` restituisce la versione corrente dell'oggetto, `photo.gif`.



Per recuperare una specifica versione occorre indicare l'ID versione. La figura seguente mostra una richiesta `GET versionId` che restituisce la versione specificata dell'oggetto (che non è necessariamente la versione corrente).



Puoi recuperare le versioni degli oggetti in Amazon S3 utilizzando la console AWS , gli SDK o l'API REST.

> **Note**
>
> Per accedere a versioni di oggetti precedenti a 300 versioni, è necessario utilizzare la AWS CLI o l'URL dell'oggetto.

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegliere il nome del bucket contenente l'oggetto.
3. Nell'elenco Oggetti scegliere il nome dell'oggetto.
4. Scegliere le Versioni.

   Amazon S3 mostra tutte le versioni per l'oggetto.
5. Selezionare la casella di controllo accanto all' ID versione per le versioni che si desidera recuperare.
6. Scegliere Azioni, scegliere Scaricae salvare l'oggetto.

È anche possibile visualizzare, scaricare ed eliminare le versioni degli oggetti nel pannello di panoramica sull'oggetto. Per ulteriori informazioni, consulta Visualizzazione della panoramica di un oggetto nella console di Amazon S3.

> **Important**
>
> È possibile annullare l'eliminazione di un oggetto solo se è stato eliminato come ultima versione (corrente). Non è possibile annullare l'eliminazione della versione precedente di un oggetto eliminato. Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Utilizzo degli SDK AWS

Gli esempi per il caricamento di oggetti in bucket senza versione e con funzione Controllo delle versioni abilitata, sono gli stessi. Tuttavia, per i bucket con funzione Controllo delle versioni abilitata, Amazon S3 assegna un numero di versione. Negli altri casi il numero di versione è null.

Per esempi di download di oggetti tramite AWS SDK per Java, .NET e PHP, consulta Download di oggetti.

Per esempi di come elencare la versione degli oggetti che utilizzano gli AWS SDK per.NET e Rust, consulta Elencare la versione degli oggetti in un bucket Amazon S3.

Utilizzo di REST API

Per recuperare una specifica versione di un oggetto

1. Impostare `versionId` sull'ID versione dell'oggetto che si desidera recuperare.
2. Inviare una richiesta `GET Object versionId`.

Example - Recupero di un oggetto con versione

La seguente richiesta recupera la versione `L4kqtJlcpXroDTDmpUMLUo` di `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=
```

È possibile recuperare solo i metadati di un oggetto (non il contenuto). Per informazioni, consulta the section called “Recupero dei metadati di una versione”.

Per informazioni sul ripristino di una versione di un oggetto precedente, consulta the section called “Ripristino di versioni precedenti”.

Recupero dei metadata di una versione di un oggetto

Se si desidera recuperare solo i metadata di un oggetto (e non il suo contenuto), si utilizza l'operazione HEAD. Per impostazione predefinita si otterranno i metadata della versione più recente. Per recuperare i metadata di una specifica versione di oggetto si indicherà l'ID versione.

Per recuperare i metadata di una versione di un oggetto

1. Impostare `versionId` sull'ID versione dell'oggetto di cui si desidera recuperare i metadata.
2. Inviare una richiesta `HEAD Object versionId`.

Example - Recupero dei metadata di un oggetto con versione

La richiesta seguente consente di recuperare i metadati della versione `3HL4kqCxf3vjVBH40Nrjfkd` di `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=
```

Di seguito è illustrata una risposta di esempio.

```
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
```

Ripristino di versioni precedenti

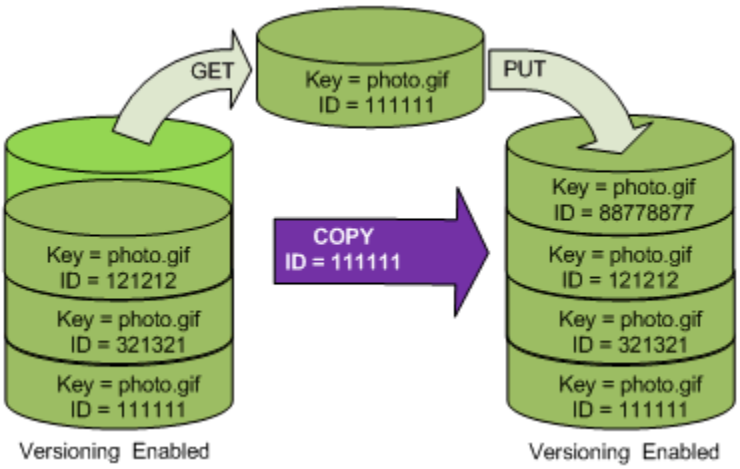
Puoi utilizzare il controllo delle versioni per recuperare le versioni precedenti di un oggetto. Esistono due metodi per farlo:

- Copiare una versione precedente dell'oggetto nello stesso bucket.

  La copia diventa la versione corrente dell'oggetto e vengono conservate tutte le sue versioni.
- Eliminare definitivamente la versione corrente dell'oggetto.

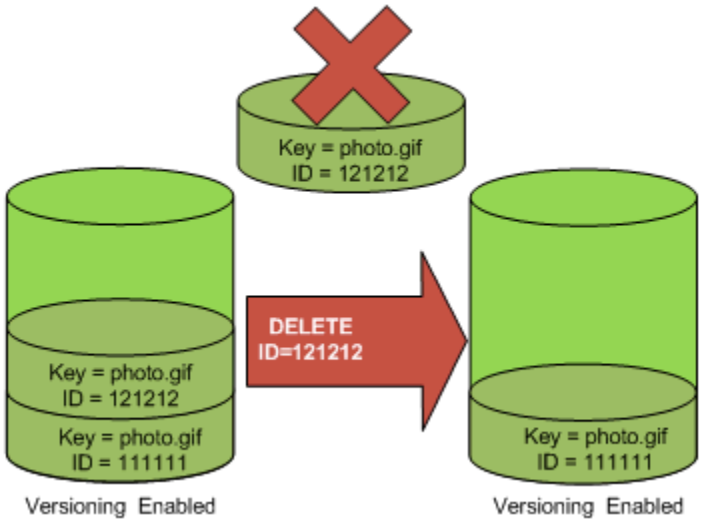
  Così facendo, in effetti, la versione precedente diventa la versione corrente dell'oggetto.

Poiché vengono mantenute tutte le versioni dell'oggetto, è possibile trasformare una qualsiasi versione precedente nella versione corrente copiano una specifica versione dell'oggetto nello stesso bucket. Nella figura seguente l'oggetto di origine (ID = 111111) viene copiato nello stesso bucket. Amazon S3 fornisce un nuovo ID (88778877), che diventa la versione corrente dell'oggetto. In questo modo il bucket conterrà sia la versione originaria dell'oggetto (111111) che la sua copia (88778877). Per ulteriori informazioni su come ottenere una versione precedente e quindi caricarla per renderla la versione corrente, consulta Recupero delle versioni degli oggetti da un bucket con funzione Controllo delle versioni abilitata e Caricamento di oggetti.



Una richiesta GET successiva recupera la versione 88778877.

La figura seguente mostra come l'eliminazione della versione corrente (121212) di un oggetto consente di lasciare la versione precedente (111111) come oggetto corrente. Per ulteriori informazioni sull'eliminazione di un oggetto, consulta Eliminazione di un singolo oggetto.



Una richiesta GET successiva recupera la versione 111111.

> **Note**
>
> Per ripristinare le versioni degli oggetti in batch, puoi utilizzare l'operazione `CopyObject`. L'operazione `CopyObject` copia ogni oggetto specificato nel manifesto. Tuttavia tieni presente che gli oggetti non vengono necessariamente copiati nello stesso ordine in cui appaiono nel manifesto. Per i bucket con versione, se è importante mantenere l'ordine di versione corrente/non corrente, è necessario copiare prima tutte le versioni non correnti. Quindi, al termine del primo processo, copia le versioni correnti in un processo successivo.

Come ripristinare le versioni precedenti degli oggetti

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegliere il nome del bucket contenente l'oggetto.
3. Nell'elenco Oggetti scegliere il nome dell'oggetto.
4. Scegliere le Versioni.

   Amazon S3 mostra tutte le versioni per l'oggetto.
5. Selezionare la casella di controllo accanto all' ID versione per le versioni che si desidera recuperare.

6. Scegliere Azioni, scegliere Scaricae salvare l'oggetto.

È anche possibile visualizzare, scaricare ed eliminare le versioni degli oggetti nel pannello di panoramica sull'oggetto. Per ulteriori informazioni, consulta Visualizzazione della panoramica di un oggetto nella console di Amazon S3.

> **Important**
>
> È possibile annullare l'eliminazione di un oggetto solo se è stato eliminato come ultima versione (corrente). Non è possibile annullare l'eliminazione della versione precedente di un oggetto eliminato. Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Utilizzo degli SDK AWS

Per informazioni sull'utilizzo di altri AWS SDK, consulta il AWS Developer Center.

Python

Il seguente esempio di codice Python ripristina la versione precedente di un oggetto con versione eliminando tutte le versioni che si sono succedute dopo la versione di rollback specificata.

```
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}."
        )
```

Eliminazione di versioni di oggetti da un bucket con funzione Controllo delle versioni abilitata

È possibile eliminare le versioni degli oggetti dai bucket Amazon S3 ogni volta che si desidera. Si possono anche definire regole di configurazione del ciclo di vita per oggetti con un ciclo di vita ben definito per fare in modo che Amazon S3 forzi la scadenza delle versioni correnti di un oggetto o che rimuova le versioni dell'oggetto non correnti in modo permanente. Se il bucket ha la funzione Controllo delle versioni abilitata o sospesa, le operazioni di configurazione del ciclo di vita agiscono nel modo seguente:

- L'operazione `Expiration` si applica alla versione corrente dell'oggetto. Anziché eliminare la versione corrente dell'oggetto, Amazon S3 la conserva come versione non corrente aggiungendo un contrassegno di eliminazione, che quindi diventa la versione corrente.
- L'operazione `NoncurrentVersionExpiration` si applica solo alle versioni non correnti di un oggetto e Amazon S3 rimuove queste versioni in modo permanente. Non è possibile ripristinare gli oggetti rimossi in modo permanente.

Per ulteriori informazioni sul ciclo di vita S3, consulta Gestione del ciclo di vita dello storage e Esempi di configurazione del ciclo di vita S3.

Per visualizzare il numero di versioni di oggetti correnti e non correnti presenti nei tuoi bucket, puoi utilizzare i parametri di Amazon S3 Storage Lens. S3 Storage Lens è una funzionalità di analisi dell'archiviazione su cloud che puoi utilizzare per avere una panoramica completa a livello di organizzazione sull'utilizzo e sulle attività relative all'archiviazione di oggetti. Per ulteriori informazioni, consulta la sezione Utilizzo di S3 Storage Lens per ottimizzare i costi di archiviazione. Per un elenco completo dei parametri, consulta Glossario dei parametri di S3 Storage.

> **Note**
>
> Le normali tariffe di Amazon S3 si applicano a ogni versione di un oggetto archiviata e trasferita, incluse le versioni non correnti dell'oggetto. Per ulteriori informazioni, consulta i Prezzi di Amazon S3.

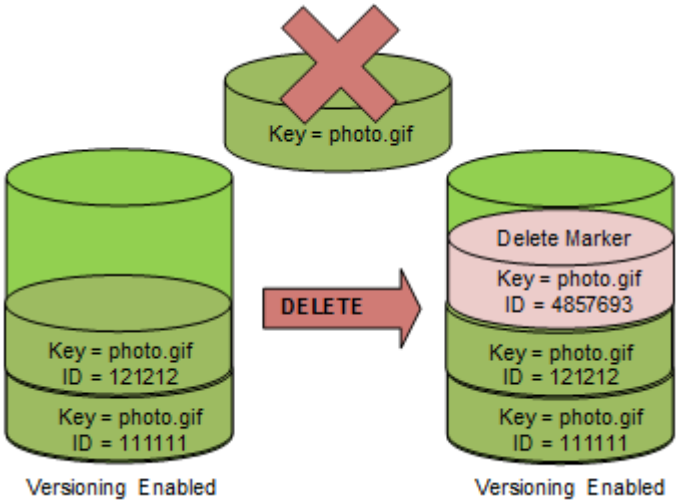
Eliminare casi di utilizzo delle richieste

Una richiesta `DELETE` può essere usata nei seguenti casi d'uso:

- Quando la funzione Controllo delle versioni è abilitata, un semplice `DELETE` non può eliminare un oggetto in modo permanente. (Una richiesta `DELETE` semplice è una richiesta che non specifica un ID versione.) Invece di eliminare l'oggetto, Amazon S3 inserisce un contrassegno di eliminazione nel bucket e tale contrassegno diventa la versione corrente dell'oggetto, con un nuovo ID.

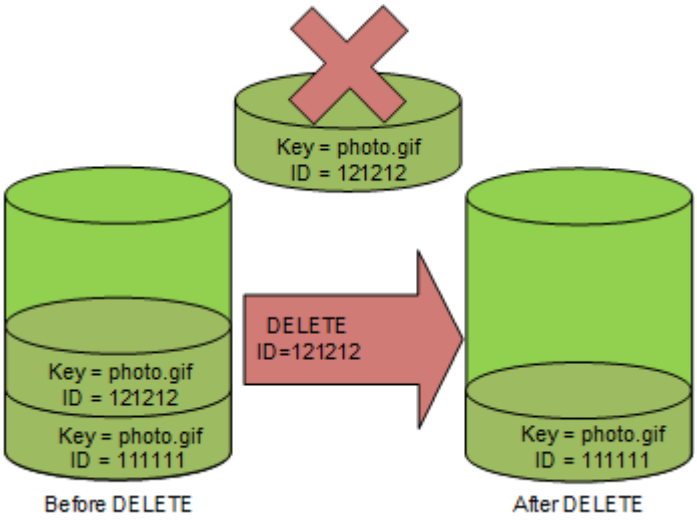
  Quando si prova a utilizzare la funzione `GET` di un oggetto la cui versione corrente è un contrassegno di eliminazione, Amazon S3 si comporta come se l'oggetto fosse stato eliminato (anche se non è stato cancellato) e restituisce un errore 404. Per ulteriori informazioni, consulta Utilizzo dei contrassegni di eliminazione.

La figura seguente mostra una richiesta DELETE semplice che non rimuove effettivamente l'oggetto specificato. Anziché rimuovere l'oggetto, Amazon S3 inserisce un contrassegno di eliminazione.



- Per eliminare oggetti con versione in modo permanente occorre usare `DELETE Object versionId`.

  La figura seguente mostra una richiesta che l'eliminazione della versione specificata di un oggetto rimuove tale oggetto in modo permanente.



Per eliminare le versioni degli oggetti

Puoi eliminare le versioni degli oggetti in Amazon S3 utilizzando la console, gli AWS SDK, l'API REST o il. AWS Command Line Interface

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegliere il nome del bucket contenente l'oggetto.
3. Nell'elenco Oggetti scegliere il nome dell'oggetto.
4. Scegliere le Versioni.

   Amazon S3 mostra tutte le versioni per l'oggetto.
5. Seleziona la casella di controllo accanto a Version ID (ID versione) per le versioni che desideri recuperare.
6. Scegliere Delete (Elimina).
7. In Eliminare definitivamente gli oggetti? , immettere **`permanently delete`**.

   > ⚠ Warning
   >
   > Quando si elimina definitivamente una versione di un oggetto, l'azione non può essere annullata.
8. Scegliere Delete objects (Elimina oggetti).

   Amazon S3 elimina la versione dell'oggetto.

Utilizzo degli SDK AWS

Per esempi di eliminazione di oggetti utilizzando gli AWS SDK per Java, .NET e PHP, consulta. Eliminazione di oggetti Amazon S3 Gli esempi per l'eliminazione di oggetti in bucket senza versione e con funzione Controllo delle versioni abilitata sono gli stessi. Tuttavia, per i bucket con funzione Controllo delle versioni abilitata, Amazon S3 assegna un numero di versione. Negli altri casi il numero di versione è null.

Per informazioni sull'utilizzo di altri AWS SDK, consulta il Developer Center.AWS

Python

Nell'esempio di codice Python seguente viene illustrata l'eliminazione permanente di un oggetto con versione eliminando tutte le sue versioni.

```
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
```

Utilizzo di REST API

Per eliminare una versione specifica di un oggetto

- In una richiesta `DELETE`, specificare l'ID versione.

Example - Eliminazione di una versione specifica

Nell'esempio seguente viene eliminata la versione `UIORUnfnd89493jJFJ` di `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
```

Utilizzando il AWS CLI

Il comando seguente elimina un oggetto denominato test.txt da un bucket denominato *`example-s3-bucket1`*. Per rimuovere una versione specifica di un oggetto, devi essere il proprietario del bucket e utilizzare la risorsa secondaria ID versione.

```
aws s3api delete-object --bucket example-s3-bucket1 --key test.txt --version-
id versionID
```

Per ulteriori informazioni su `delete-object`, consulta delete-object nel Riferimento ai comandi AWS CLI .

Per ulteriori informazioni sull'eliminazione delle versioni degli oggetti, consulta gli argomenti riportati di seguito.

- Utilizzo dei contrassegni di eliminazione
- Rimozione dei contrassegni di eliminazione per rendere corrente una versione precedente
- Eliminazione di un oggetto da un bucket con cancellazione MFA abilitata

Utilizzo dei contrassegni di eliminazione

In Amazon S3, un contrassegno di eliminazione è il segnaposto (o contrassegno) di un oggetto con controllo delle versioni specificato in una richiesta `DELETE` semplice. Una richiesta `DELETE` semplice è una richiesta che non specifica un ID versione. Poiché l'oggetto si trova in un bucket con funzione Controllo delle versioni abilitata, non viene eliminato. Ma il contrassegno di eliminazione fa sì che Amazon S3 si comporti come se l'oggetto fosse stato eliminato. Puoi utilizzare una chiamata `DELETE` API di Amazon S3 su un contrassegno di eliminazione. A tale scopo, è necessario effettuare la `DELETE` richiesta utilizzando un utente o un ruolo AWS Identity and Access Management (IAM) con le autorizzazioni appropriate.

I contrassegni di eliminazione sono dotati di un nome chiave (o chiave) e di un ID versione, come qualsiasi altro oggetto. Tuttavia, differiscono da altri oggetti nei modi seguenti:

- Un contrassegno di eliminazione non dispone di dati associati.
- Un contrassegno di eliminazione non è associato a un valore della lista di controllo degli accessi (ACL).
- Se invii una richiesta `GET` per un contrassegno di eliminazione, la richiesta `GET` non recupera nulla perché un contrassegno di eliminazione non contiene dati. In particolare, quando la richiesta `GET` non specifica un `versionId`, viene visualizzato un errore 404 (Not Found).

I contrassegni di eliminazione accumulano un addebito minimo per l'archiviazione in Amazon S3. Le dimensioni di storage di un contrassegno di eliminazione corrispondono a quelle del suo nome delle chiave. Un nome delle chiave è una sequenza di caratteri Unicode. La codifica UTF-8 per il nome

chiave aggiunge da 1 a 4 byte di archiviazione al bucket per ogni carattere contenuto nel nome. I contrassegni di eliminazione sono archiviati nella classe di archiviazione S3 Standard.

Per scoprire quanti contrassegni di eliminazione sono impostati e in quale classe di archiviazione sono archiviati, puoi usare Amazon S3 Storage Lens. Per ulteriori informazioni, consulta Valutazione dell'attività e dell'utilizzo dello storage con Amazon S3 Storage Lens e Glossario dei parametri di Amazon S3 Storage Lens.

Per ulteriori informazioni sui nomi delle chiavi, consultare Creazione di nomi di chiavi oggetto. Per informazioni sull'eliminazione di un contrassegno di eliminazione, consultare Gestione dei contrassegni di eliminazione.

Solo Amazon S3 può creare un contrassegno di eliminazione e compie questa operazione ogni volta che si invia una richiesta `DeleteObject` relativa a un oggetto di un bucket con funzione Controllo delle versioni abilitata o sospesa. L'oggetto specificato nella richiesta `DELETE` non viene effettivamente eliminato. Invece il contrassegno di eliminazione diventa la versione corrente dell'oggetto Il nome delle chiave dell'oggetto (o chiave) diventa la chiave del contrassegno di eliminazione.

Quando ottieni un oggetto senza specificare un `versionId` nella richiesta, se la versione corrente è un contrassegno di eliminazione, Amazon S3 risponde con quanto segue:

- Un errore 404 (Not Found)
- Un'intestazione di risposta, `x-amz-delete-marker: true`

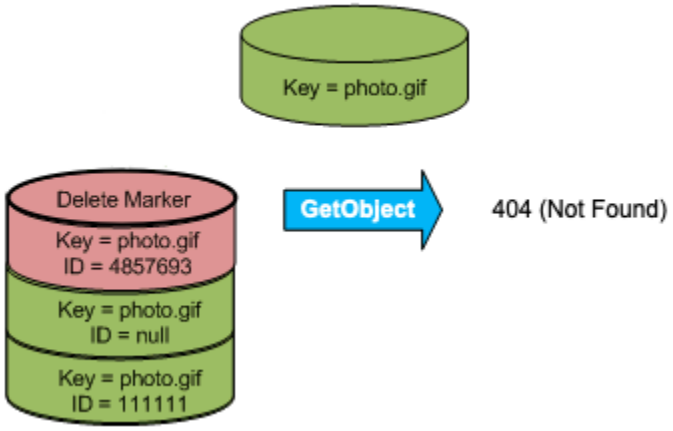
Quando ottieni un oggetto specificando un `versionId` nella richiesta, se la versione specificata è un contrassegno di eliminazione, Amazon S3 risponde con quanto segue:

- Un errore di tipo 405 (metodo non concesso)
- Un'intestazione di risposta, `x-amz-delete-marker: true`
- Un'intestazione di risposta, `Last-Modified: timestamp` (solo quando si utilizzano le operazioni HeadObjecto l'GetObjectAPI)

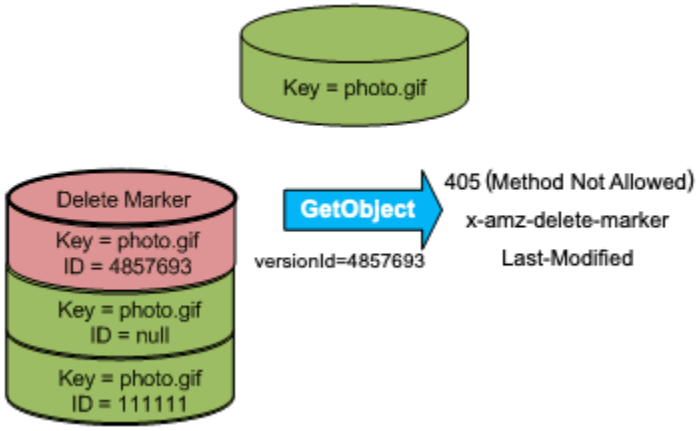
L'intestazione della risposta `x-amz-delete-marker: true` indica che l'oggetto a cui è stato effettuato l'accesso è un contrassegno di eliminazione. Questa intestazione della risposta non restituisce mai `false`, perché quando il valore è `false`, la versione corrente o specificata dell'oggetto non è un indicatore di eliminazione.

L'intestazione della risposta `Last-Modified` fornisce l'ora di creazione dei contrassegni di eliminazione.

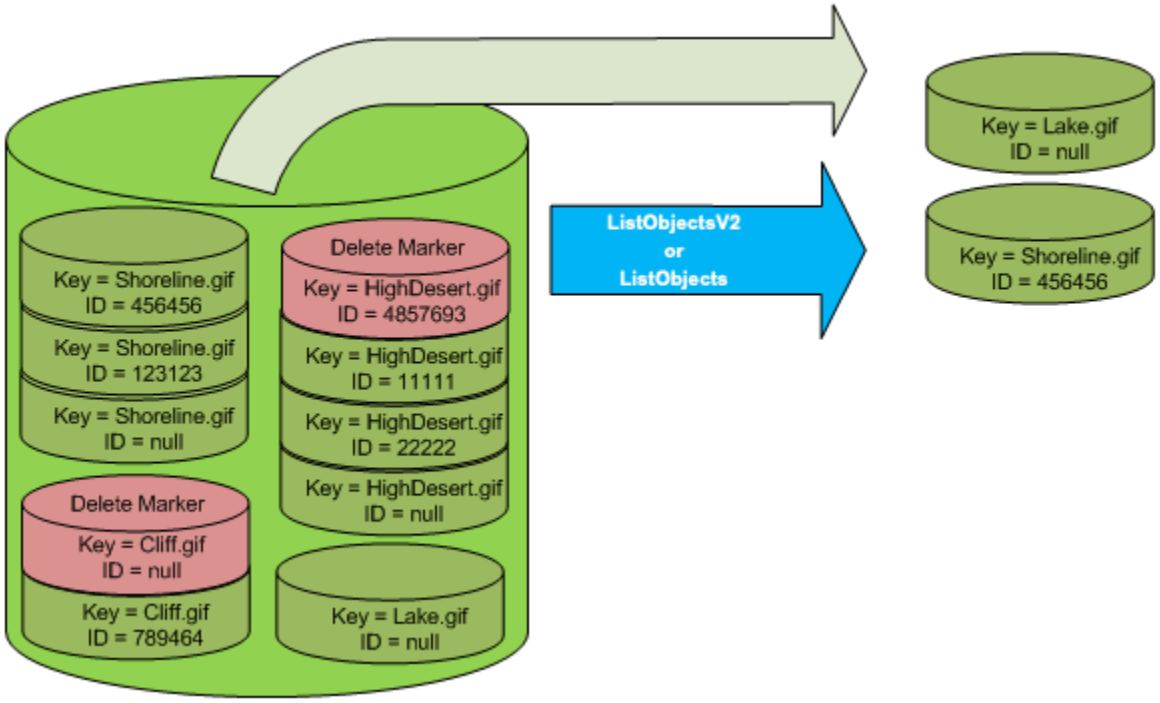
La figura seguente mostra come una chiamata API `GetObject` su un oggetto la cui versione corrente è un contrassegno di eliminazione risponde con un errore 404 (Not Found) e l'intestazione della risposta include `x-amz-delete-marker: true`.



Se effettui una chiamata `GetObject` su un oggetto specificando un `versionId` nella richiesta e se la versione specificata è un contrassegno di eliminazione, Amazon S3 risponde con un errore 405 (Method Not Allowed) e le intestazioni della risposta includono `x-amz-delete-marker: true` e `Last-Modified: timestamp`.



L'unico modo per ottenere un elenco dei contrassegni di eliminazione (e di altre versioni di un oggetto) è utilizzare la sottorisorsa `versions` in una richiesta ListObjectVersions. La figura seguente mostra che una richiesta ListObjectsV2 o ListObjects non restituisce gli oggetti la cui versione corrente è un contrassegno di eliminazione.



Gestione dei contrassegni di eliminazione

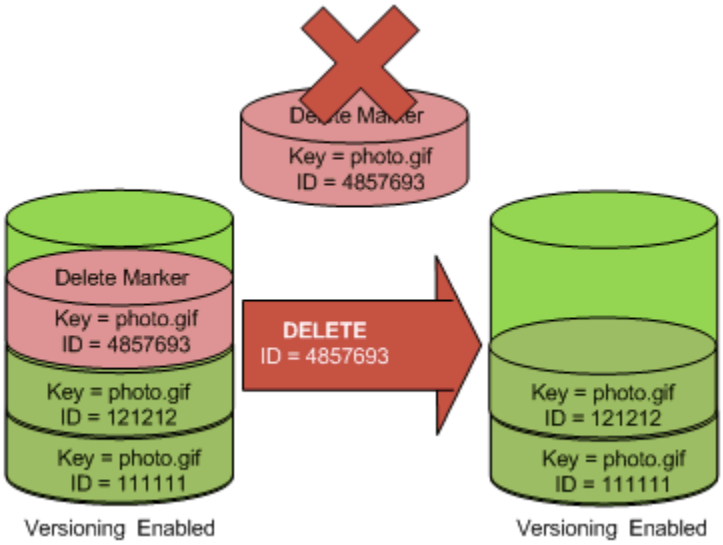
Configurazione del ciclo di vita per ripulire automaticamente i contrassegni di eliminazione scaduti

Un contrassegno di eliminazione oggetto scaduto è un elemento in cui tutte le versioni dell'oggetto vengono eliminate e rimane solo un singolo contrassegno di eliminazione. Se la configurazione relativa al ciclo di vita è impostata per eliminare le versioni correnti oppure l'opzione `ExpiredObjectDeleteMarker` è impostata in modo esplicito, Amazon S3 rimuove il contrassegno di eliminazione dell'oggetto scaduto. Per un esempio, consulta Esempio 7: rimozione dei contrassegni di eliminazione oggetto scaduto.

Rimozione dei contrassegni di eliminazione per rendere corrente una versione precedente

Quando si elimina un oggetto in un bucket che supporta la funzione Controllo delle versioni, tutte le versioni rimangono nel bucket e Amazon S3 crea un contrassegno di eliminazione per l'oggetto. Per annullare l'eliminazione dell'oggetto, è necessario eliminare il contrassegno di eliminazione. Per ulteriori informazioni sulla funzione Controllo delle versioni e sui contrassegni di eliminazione, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

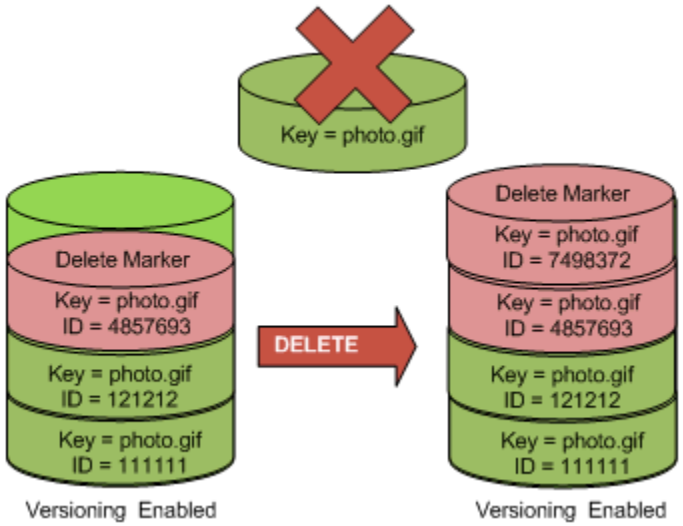
Per eliminare definitivamente un contrassegno di eliminazione occorre includere il suo ID versione nella richiesta `DeleteObject versionId`. La figura seguente mostra una richiesta `DeleteObject versionId` che rimuove definitivamente un contrassegno di eliminazione.



L'effetto della rimozione del contrassegno di eliminazione è che una richiesta `GET` semplice non recupererà l'ID versione corrente (121212) dell'oggetto.

> **Note**
>
> Se si utilizza una richiesta `DeleteObject` per eliminare un contrassegno di eliminazione (senza specificare l'ID versione del contrassegno), Amazon S3 non elimina il contrassegno, ma `PUT`s inserisce un altro contrassegno di eliminazione.

Per rimuovere un contrassegno di eliminazione con un ID di versione `NULL`, è necessario passare il `NULL` come ID di versione nella richiesta `DeleteObject`. La figura seguente mostra come una semplice richiesta `DeleteObject` effettuata senza un ID di versione, in cui la versione corrente è un marker di eliminazione, non rimuove nulla, ma aggiunge invece un marker di eliminazione ulteriore con un ID di versione univoco (7498372).



Utilizzo della console S3

Utilizzare la seguente procedura per recuperare gli oggetti eliminati che non sono cartelle dal bucket S3, inclusi gli oggetti che si trovano all'interno di tali cartelle.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket scegli il nome del bucket desiderato.
3. Per visualizzare un elenco delle versioni degli oggetti nel bucket, scegliere l'opzione List versions (Elenca versioni). Verranno visualizzati i contrassegni di eliminazione degli oggetti eliminati.
4. Per annullare l'eliminazione di un oggetto, è necessario eliminare il contrassegno di eliminazione. Selezionare la casella di controllo accanto al contrassegno di eliminazione dell'oggetto da recuperare, quindi scegliere Delete (Elimina).
5. Conferma l'eliminazione nella pagina Delete objects (Elimina oggetti) .
   a. In Permanently delete objects? (Eliminare definitivamente gli oggetti?), specifica **`permanently delete`**.
   b. Scegliere Delete objects (Elimina oggetti).

> **Note**
>
> Non puoi utilizzare la console di Amazon S3 per annullare l'eliminazione delle cartelle. È necessario utilizzare il AWS CLI o SDK. Per gli esempi, consulta Come posso ripristinare

un oggetto Amazon S3 eliminato da un bucket con il controllo delle versioni abilitato? nel Knowledge Center di AWS .

Utilizzo di REST API

Per rimuovere definitivamente un contrassegno di eliminazione

1. Impostare `versionId` sull'ID versione del contrassegno di eliminazione che si desidera rimuovere.
2. Inviare una richiesta `DELETE Object versionId`.

Example - Rimozione di un contrassegno di eliminazione

Il seguente esempio consente di rimuovere il contrassegno di eliminazione della versione budget 4857693 di `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=
```

Quando si elimina un contrassegno di eliminazione, Amazon S3 include nella risposta:

```
204 NoContent
x-amz-version-id: versionID
x-amz-delete-marker: true
```

Utilizzo degli SDK AWS

Per informazioni sull'utilizzo di altri AWS SDK, consulta il AWS Developer Center.

Python

Nell'esempio di codice Python seguente viene illustrato come rimuovere un marker di eliminazione da un oggetto, rendendo quindi la versione non corrente più recente la versione corrente dell'oggetto.

```
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)
```

Eliminazione di un oggetto da un bucket con cancellazione MFA abilitata

Se la configurazione della funzione Controllo delle versioni comprende l'abilitazione della cancellazione MFA, il proprietario del bucket deve includere l'intestazione `x-amz-mfa` nelle richieste per eliminare definitivamente una versione dell'oggetto o per cambiare lo stato della funzione Controllo delle versioni del bucket. Le richieste che includono `x-amz-mfa` devono utilizzare l'HTTPS.

Il valore dell'intestazione è dato dalla concatenazione del numero di serie del dispositivo di autenticazione, uno spazio e il codice di autenticazione visualizzato sul dispositivo. Se non si include questa intestazione di richiesta, la richiesta ha esito negativo.

Per ulteriori informazioni sui dispositivi di autenticazione, vedere Autenticazione a più fattori.

Example - Eliminazione di un oggetto da un bucket con cancellazione MFA abilitata

L'esempio seguente mostra come eliminare `my-image.jpg` (con la versione specificata), che risiede in un bucket configurato con l'eliminazione MFA abilitata.

Nota lo spazio tra *`[SerialNumber]`* e *`[AuthenticationCode]`*. Per ulteriori informazioni, consulta DeleteObject in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di 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=
```

Per ulteriori informazioni sull'abilitazione della cancellazione MFA;, consultare Configurazione dell'eliminazione di MFA.

Configurazione delle autorizzazioni degli oggetti con versione

Le autorizzazioni per gli oggetti in Amazon S3 sono impostate a livello di versione. Ogni versione ha il proprio proprietario dell'oggetto. Chi crea Account AWS la versione dell'oggetto è il proprietario. È quindi possibile definire autorizzazioni diverse per versioni differenti dello stesso oggetto. A tale scopo occorre specificare l'ID versione dell'oggetto le cui autorizzazioni si desidera impostare in una

richiesta `PUT Object versionId acl`. Per una descrizione dettagliata e per le istruzioni di utilizzo delle ACL, consultare Identity and Access Management per Amazon S3.

Example - Configurazione delle autorizzazioni di un oggetto con versione

La richiesta seguente consente di impostare l'autorizzazione dell'assegnatario, `BucketOwner@amazon.com`, su `FULL_CONTROL` per la chiave, `my-image.jpg`, ID versione, `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>
```

Analogamente, per conoscere le autorizzazioni della versione specifica di un oggetto, è necessario indicarne l'ID versione in una richiesta `GET Object versionId acl`. Includere l'ID versione è necessario perché, per impostazione predefinita, `GET Object acl` restituisce le autorizzazioni della versione corrente dell'oggetto.

Example - Recupero delle autorizzazioni della versione specificata di un oggetto

Nell'esempio seguente Amazon S3 restituisce le autorizzazioni per la chiave, `my-image.jpg`, ID versione, `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
```

Per ulteriori informazioni, consulta GetObjectAcl in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Utilizzo di oggetti di un bucket con funzione Controllo delle versioni sospesa

In Amazon S3 è possibile sospendere la funzione Controllo delle versioni per non accumulare nuove versioni dello stesso oggetto in un bucket. Potrebbe essere necessario farlo perché si desidera solo una singola versione di un oggetto in un bucket. In alternativa, potrebbe esservi la necessità di non voler accumulare addebiti per più versioni.

Quando si sospende la funzione Controllo delle versioni, gli oggetti esistenti nel bucket non si modificano. Ciò che cambia è il modo in cui Amazon S3 gestirà gli oggetti delle richieste future. Negli argomenti di questa sezione vengono illustrate le varie operazioni degli oggetti in un bucket con la funzione Controllo delle versioni sospesa, tra cui l'aggiunta, il recupero e l'eliminazione di oggetti.

Per ulteriori informazioni sulla funzione Controllo delle versioni S3, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3. Per ulteriori informazioni sul recupero delle versioni degli oggetti, consulta la sezione Recupero delle versioni degli oggetti da un bucket con funzione Controllo delle versioni abilitata.

Argomenti

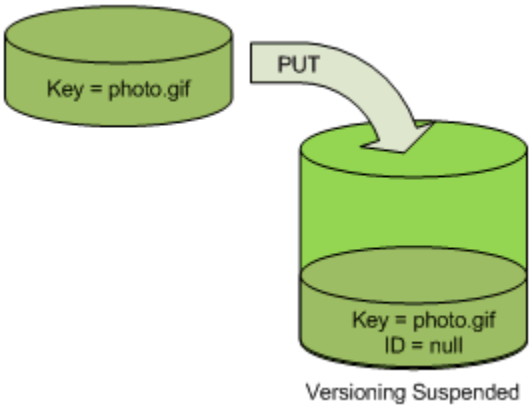
- Aggiunta di oggetti a bucket con funzione Controllo delle versioni sospesa
- Recupero di oggetti da bucket con funzioneControllo delle versioni sospesa
- Eliminazione di oggetti da bucket con funzione Controllo delle versioni sospesa

Aggiunta di oggetti a bucket con funzione Controllo delle versioni sospesa

Puoi aggiungere oggetti a bucket con la funzione Controllo delle versioni sospesa in Amazon S3 per creare l'oggetto con ID versione null o sovrascrivere una qualsiasi versione dell'oggetto con un ID versione corrispondente.

Dopo la sospensione della funzione Controllo delle versioni di un bucket, Amazon S3 aggiungerà automaticamente un ID versione `null` a ogni oggetto archiviato successivamente (utilizzando `PUT`, `POST` o `CopyObject`) nel bucket.

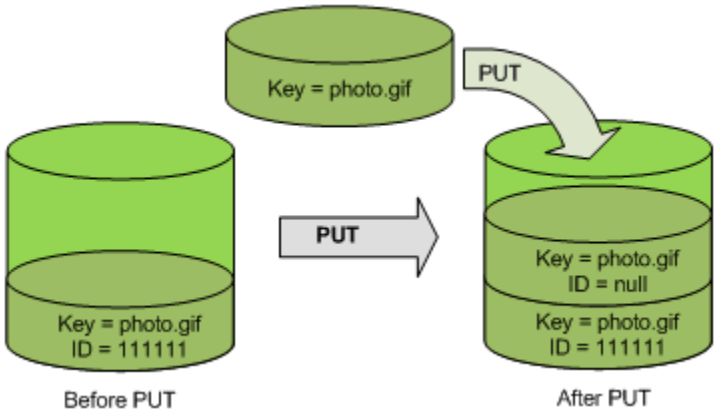
La figura seguente mostra l'aggiunta dell'ID versione `null` a un oggetto da parte di Amazon S3 quando l'oggetto viene aggiunto a un bucket con funzione Controllo delle versioni abilitata.



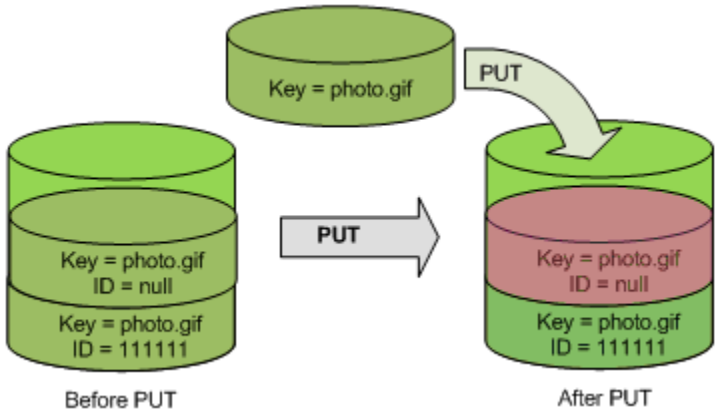
Se nel bucket è già presente una versione null e si aggiunge un altro oggetto con la stessa chiave, l'oggetto così aggiunto sovrascrive la versione null originaria.

Se il bucket contiene oggetti con versione, la versione della funzione `PUT` diventa quella corrente dell'oggetto. La figura seguente mostra come l'aggiunta di un oggetto a un bucket contenente oggetti con versione non sovrascrive l'oggetto già presente nel bucket.

In questo caso, la versione 111111 risiedeva già nel bucket. Amazon S3 aggiunge un ID versione null all'oggetto da aggiungere e lo archivia nel bucket. La versione 111111 risulta ora sovrascritta.



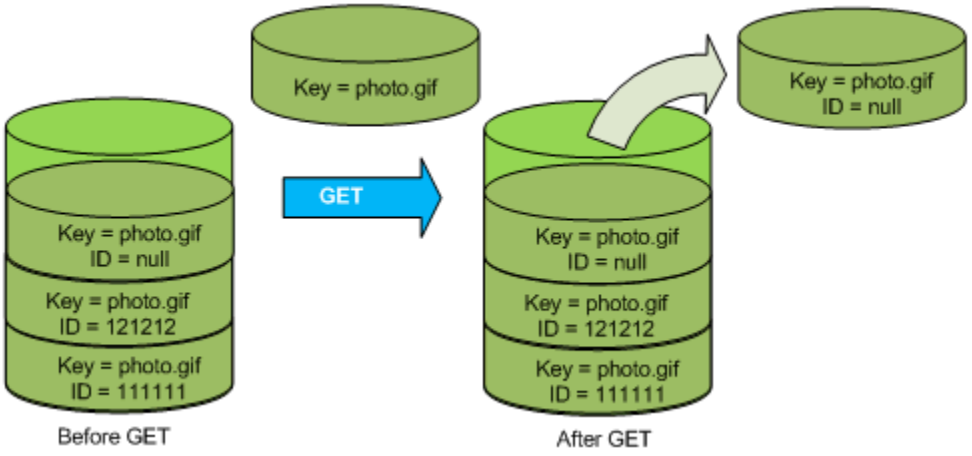
Se nel bucket è già presente una versione null, tale versione viene sovrascritta, come mostrato nell'illustrazione seguente.



Sebbene la chiave e l'ID (`null`) della versione null siano identici prima e dopo la richiesta PUT, i contenuti della versione null inizialmente memorizzati nel bucket vengono sostituiti da quelli dell'oggetto PUT per l'inserimento nel bucket.

Recupero di oggetti da bucket con funzioneControllo delle versioni sospesa

Le richieste `GET Object` restituiscono la versione corrente di un oggetto indipendentemente dal fatto che la funzione Controllo delle versioni del bucket sia stata abilitata o meno. La figura seguente mostra come un semplice `GET` restituisce la versione corrente di un oggetto.



Eliminazione di oggetti da bucket con funzione Controllo delle versioni sospesa

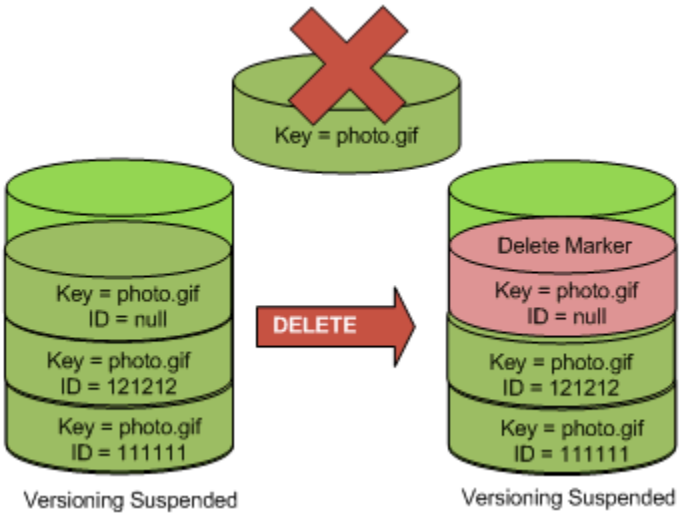
È possibile eliminare oggetti da bucket con la funzione Controllo delle versioni sospesa per rimuovere un oggetto con ID versione null.

Se la funzione Controllo delle versioni è sospesa per un bucket, una richiesta `DELETE`:

- Può rimuovere solo gli oggetti con ID versione `null`.
- Non rimuove alcun oggetto se non è presente una versione null dell'oggetto nel bucket.

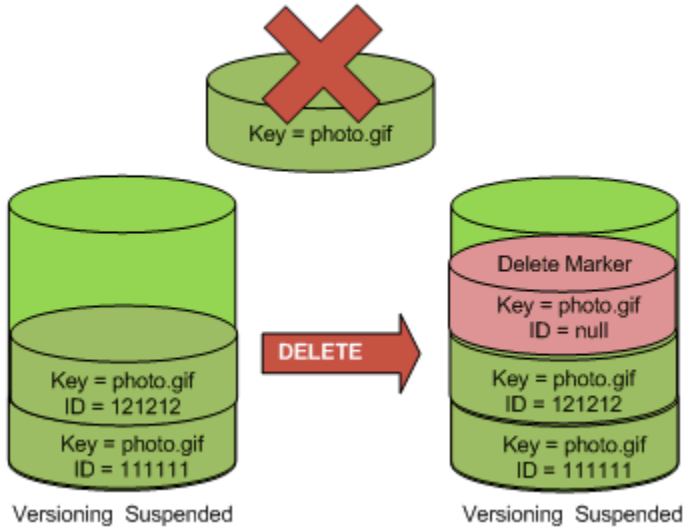
- Inserisce un contrassegno di eliminazione nel bucket.

La figura seguente mostra come un semplice operazione `DELETE` rimuove una versione null. (Una richiesta `DELETE` semplice è una richiesta che non specifica un ID versione.) Amazon S3 inserisce un contrassegno di eliminazione al suo posto con un ID versione di `null`.



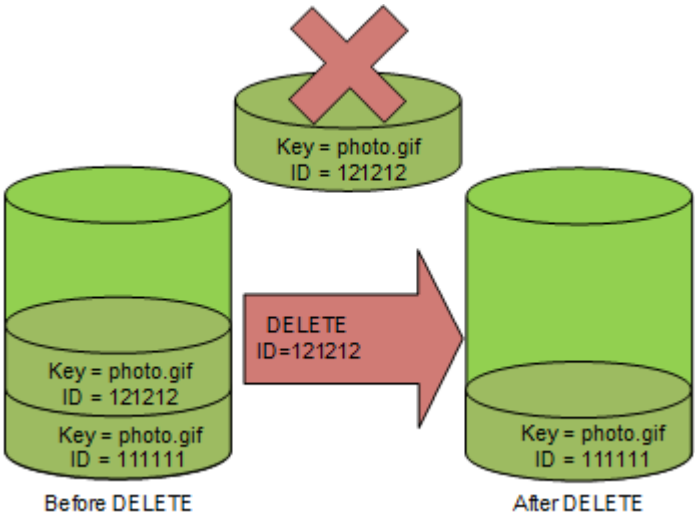
Ricordare che i contrassegni di eliminazione non hanno contenuto, pertanto il contenuto della versione null viene perso quando è sostituito da un contrassegno di eliminazione.

La figura seguente mostra un bucket che non contiene versioni null. In questo caso la richiesta `DELETE` non rimuove nulla. Amazon S3 si limita a inserire un contrassegno di eliminazione.



Anche nei bucket con funzione di controllo delle versioni sospesa, il proprietario del bucket può eliminare definitivamente la versione specificata includendo l'ID versione nella richiesta `DELETE`.

La figura seguente mostra che l'eliminazione di una versione specificata di un oggetto rimuove tale oggetto in modo permanente. Solo il proprietario del bucket può eliminare la versione specificata di un oggetto.



Utilizzo di AWS Backup per Amazon S3

Amazon S3 è integrato in modo nativo con AWS Backup, un servizio completamente gestito e basato su policy che puoi utilizzare per definire una policy di backup centrale per proteggere i tuoi dati in Amazon S3. Dopo aver definito le policy di backup e assegnato le risorse Amazon S3 alle policy, AWS Backup automatizza la creazione di backup di Amazon S3 e archivia in modo sicuro i backup nel vault di backup crittografato che hai designato nel piano di backup.

Quando si utilizza AWS Backup per Amazon S3, puoi eseguire le seguenti operazioni:

- Creare backup continui e backup periodici. I backup continui sono utili per il ripristino point-in-time e i backup periodici sono utili per soddisfare le esigenze di conservazione dei dati a lungo termine.
- Automatizzare la pianificazione e conservazione dei backup configurando centralmente le policy di backup.
- Ripristinare i backup dei dati Amazon S3 di un momento specifico.

Con AWS Backup, puoi utilizzare il controllo delle versioni S3 e la replica S3 per recuperare facilmente i dati da eliminazioni accidentali ed eseguire operazioni di auto-ripristino.

Prerequisiti

È necessario attivare S3 Versioning (Controllo delle versioni S3) per il bucket per poterne eseguire il backup con AWS Backup.

> **Note**
>
> Ti consigliamo di impostare una regola di scadenza del ciclo di vita per i bucket con il controllo delle versioni abilitato che vengono sottoposti a backup. Se non imposti un periodo di scadenza del ciclo di vita, i costi di archiviazione di Amazon S3 potrebbero aumentare perché AWS Backup mantiene tutte le versioni dei dati di Amazon S3.

Nozioni di base

Per iniziare a utilizzare AWS Backup per Amazon S3, consulta Creating Amazon S3 backups (Creazione di backup di Amazon S3) nella Guida per gli sviluppatori di AWS Backup.

Restrizioni e limitazioni

Per informazioni sulle limitazioni, consulta Creating Amazon S3 backups (Creazione di backup di Amazon S3) nella Guida per gli sviluppatori di AWS Backup.

Utilizzo di oggetti archiviati

Per ridurre i costi di archiviazione degli oggetti a cui si accede raramente, è possibile archiviare tali oggetti. Quando si archivia un oggetto, questo viene spostato in una archiviazione a basso costo, il che significa che non è possibile accedervi in tempo reale.

Sebbene gli oggetti archiviati non siano accessibili in tempo reale, è possibile ripristinarli in pochi minuti o ore, a seconda della classe di archiviazione. Puoi ripristinare un oggetto archiviato utilizzando la console Amazon S3, S3 Batch Operations, l'API REST, AWS gli SDK e (). AWS Command Line Interface AWS CLI Per istruzioni, consulta Ripristino di un oggetto archiviato.

Gli oggetti Amazon S3 nelle classi o nei livelli di archiviazione seguenti sono archiviati e non sono accessibili in tempo reale:

- Classe di archiviazione S3 Glacier Flexible Retrieval (Recupero flessibile S3 Glacier)
- Classe di archiviazione S3 Glacier Deep Archive (Archiviazione profonda S3 Glacier)
- Livello S3 Intelligent-Tiering Archive Access (Accesso archiviazione Piano intelligente S3)
- Livello Deep Archive Access di Piano intelligente Amazon S3

Per ripristinare gli oggetti, è necessario completare le seguenti operazioni:

- Per gli oggetti nelle classi di archiviazione S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, è necessario avviare una richiesta di ripristino e quindi attendere fino a quando non sia disponibile una copia temporanea dell'oggetto. Quando viene creata una copia temporanea dell'oggetto ripristinato, la classe di archiviazione dell'oggetto rimane la stessa. Una richiesta di operazioni API HeadObject o GetObject ripristina la classe di archiviazione S3 Glacier Flexible Retrieval (Recupero flessibile S3 Glacier) o S3 Glacier Deep Archive (Archiviazione profonda S3 Glacier).
- Per gli oggetti nei livelli Accesso archivio di S3 Intelligent-Tiering e Accesso archivio approfondito di S3 Intelligent-Tiering, è innanzitutto necessario avviare una richiesta di ripristino e quindi attendere che l'oggetto venga spostato nel livello Accesso frequente.

Per maggiori informazioni sul confronto di tutte le classi di storage di Amazon S3, consulta Utilizzo delle classi di storage di Amazon S3. Per ulteriori informazioni su S3 Intelligent-Tiering (Piano intelligente S3), consulta the section called “Come funziona S3 Intelligent-Tiering”.

Ripristino di oggetti da S3 Glacier

Quando si utilizza S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, Amazon S3 ripristina una copia temporanea dell'oggetto solo per la durata specificata. Successivamente, elimina la copia ripristinata dell'oggetto. Puoi modificare il periodo di scadenza di una copia ripristinata eseguendo nuovamente una richiesta di ripristino. In questo caso, Amazon S3 aggiorna il periodo di scadenza relativo all'ora corrente.

> **Note**
>
> Quando ripristini un oggetto archiviato da S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, paghi sia per l'oggetto archiviato che per una copia temporaneamente ripristinata. Per informazioni sui prezzi, consulta Prezzi di Amazon S3.

Ripristino degli oggetti da S3 Intelligent-Tiering

Quando si esegue il ripristino dal livello di Accesso archivio di S3 Intelligent-Tiering o Accesso archivio approfondito di S3 Intelligent-Tiering, l'oggetto torna al livello S3 Intelligent-Tiering Frequent Access. In seguito, se non si accede all'oggetto per 30 giorni consecutivi, l'oggetto viene spostato automaticamente nel livello Infrequent Access (Accesso Infrequente). L'oggetto passa

automaticamente al livello S3 Intelligent-Tiering Archive Access (Accesso archiviazione Piano intelligente Amazon S3) dopo un minimo di 90 giorni consecutivi senza accesso. Se non si accede all'oggetto dopo un minimo di 180 giorni consecutivi, l'oggetto passa al livello Deep Archive Access (Accesso archiviazione profonda).

> **Note**
>
> A differenza delle classi di archiviazione di S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive, le richieste di ripristino per gli oggetti S3 Intelligent-Tiering non accettano il valore `Days`.

Utilizzo di Operazioni in batch S3 con richieste di ripristino

Per ripristinare più di un oggetto Amazon S3 con una sola richiesta, è possibile utilizzare le operazioni in batch S3. Fornisci alle operazioni in batch S3 un elenco di oggetti su cui operare. Le operazioni in batch S3 richiamano la rispettiva API per eseguire l'operazione specificata. Un solo processo di operazioni in batch può eseguire l'operazione specificata su miliardi di oggetti contenenti esabyte di dati.

Tempo di ripristino

Amazon S3 calcola l'ora di scadenza della copia dell'oggetto ripristinato aggiungendo il numero di giorni specificati nella richiesta di ripristino all'ora in cui è stato completato il ripristino richiesto. Amazon S3 arrotonda quindi l'ora risultante alla mezzanotte UTC (Universal Coordinated Time) del giorno successivo. Ad esempio, supponiamo che la copia di un oggetto ripristinato sia stata creata il 15 ottobre 2012, alle 10:30 AM UTC e che come periodo di ripristino sia stato specificato un valore di tre giorni. In questo caso, la copia ripristinata scade il 19 ottobre 2012, 00:00 UTC; a quel punto Amazon S3 elimina la copia dell'oggetto.

Il tempo necessario per completare un processo di ripristino dipende dalla classe o dal livello di archiviazione utilizzato e dall'opzione di recupero specificata: Expedited (disponibile solo per S3 Glacier Flexible Retrieval e Accesso archivio di S3 Intelligent-Tiering), Standard, o Bulk. Per ulteriori informazioni, consulta Opzioni di recupero dall'archivio.

Puoi ricevere una notifica quando il ripristino viene completato mediante la funzionalità di notifica eventi di Amazon S3. Per ulteriori informazioni, consulta Notifiche di eventi Amazon S3.

Argomenti

Opzioni di recupero dall'archivio

Di seguito sono elencate le opzioni di recupero disponibili per ripristinare un oggetto archiviato in Amazon S3:

- Expedited: accesso rapido ai dati archiviati nella classe di archiviazione S3 Glacier Flexible Retrieval o nel livello Accesso archivio di S3 Intelligent-Tiering. È possibile utilizzare questa opzione quando sono necessarie richieste urgenti occasionali per un sottoinsieme di archivi. Per tutti gli oggetti archiviati ad eccezione dei più voluminosi (oltre 250 MB), i dati ai quali è possibile accedere tramite i recuperi velocizzati sono disponibili generalmente entro 1–5 minuti.

  > **Note**
  >
  > I recuperi rapidi sono una funzionalità premium e vengono addebitati in base alla tariffa di richiesta e recupero rapidi.
  > Per informazioni sui prezzi di Amazon S3, consulta Prezzi di Amazon S3.

  La capacità assegnata assicura che la capacità di recupero per effettuare recuperi di tipo rapido da S3 Glacier Flexible Retrieval sia disponibile in base alla necessità. Per ulteriori informazioni, consulta Capacità con provisioning.
- Standard: accesso a qualsiasi oggetto archiviato entro alcune ore. Questa è l'opzione predefinita per le richieste di recupero che non specificano l'opzione di recupero. I recuperi standard in genere terminano entro 3-5 ore per gli oggetti archiviati nella classe di archiviazione S3 Glacier Flexible Retrieval o nel livello S3 Intelligent-Tiering Archive Access. Questi recuperi in genere terminano entro 48 ore per gli oggetti archiviati nella classe di archiviazione S3 Glacier Deep Archive (Archiviazione profonda S3 Glacier) o nel livello S3 Intelligent-Tiering Deep Archive Access (Accesso archiviazione profonda Piano intelligente S3). I recuperi standard sono gratuiti per gli oggetti archiviati nel Piano intelligente Amazon S3.

  > **Note**
  >
  > - Per gli oggetti archiviati nella classe di storage S3 Glacier Flexible Retrieval o nel livello S3 Intelligent-Tiering Archive Access, i recuperi standard avviati utilizzando l'operazione

di ripristino di S3 Batch Operations iniziano in genere entro pochi minuti e terminano entro 3-5 ore.

- Per gli oggetti nella classe di storage S3 Glacier Deep Archive o nel livello S3 Intelligent-Tiering Deep Archive Access, i recuperi Standard avviati utilizzando l'operazione di ripristino Batch Operations iniziano in genere entro 9 ore e terminano entro 12 ore.

- Bulk: accede ai dati usando l'opzione di recupero più economica in Amazon S3 Glacier. I recuperi Bulk consentono di recuperare grandi quantità di dati, fino a petabyte, in modo conveniente.

  Per gli oggetti archiviati nella classe di storage S3 Glacier Flexible Retrieval o nel livello S3 Intelligent-Tiering Archive Access, i recuperi in blocco in genere terminano entro 5-12 ore. Per gli oggetti archiviati nella classe di storage S3 Glacier Deep Archive o nel livello S3 Intelligent-Tiering Deep Archive Access, questi recuperi in genere terminano entro 48 ore.

  I recuperi in blocco sono gratuiti per gli oggetti archiviati nelle classi di storage S3 Glacier Flexible Retrieval o S3 Intelligent-Tiering.

La tabella seguente riepiloga le opzioni di recupero archivi. Per informazioni sui prezzi, consulta Prezzi di Amazon S3.

Per creare un`Expedited`, o `Bulk` recuperarlo`Standard`, imposta l'elemento di richiesta nella `Tier` richiesta di operazione API RestoreObjectREST sull'opzione desiderata o sull'equivalente in () o SDK. AWS Command Line Interface AWS CLI AWS Se hai acquistato capacità con provisioning, tutti i recuperi Expedited vengono serviti automaticamente mediante la capacità con provisioning.

Capacità con provisioning

La capacità con provisioning assicura che la capacità di recupero per effettuare recuperi di tipo Expedited da S3 Glacier Flexible Retrieval sia disponibile in base alla necessità. Ogni unità di capacità assicura almeno tre recuperi di tipo Expedited ogni 5 minuti e fornisce fino a 150 MB/s di velocità di trasmissione effettiva per il recupero.

Se il carico di lavoro richiede un accesso altamente affidabile e prevedibile a un sottoinsieme di dati nell'arco di pochi minuti, è necessario acquistare capacità di recupero con provisioning. Senza capacità con provisioning, i recuperi Expedited potrebbero non essere accettati durante periodi di richiesta elevata. Se è necessario l'accesso ai recuperi Expedited in qualsiasi circostanza, è consigliabile acquistare capacità di recupero assegnata.

Le unità di capacità assegnate vengono assegnate a un. Account AWS Pertanto, il richiedente del recupero Expedited dei dati dovrebbe acquistare l'unità di capacità assegnata, non il proprietario del bucket.

Puoi acquistare la capacità fornita utilizzando la console Amazon S3, la console Amazon S3 Glacier, l'operazione API REST Purchase Provisioned Capacity, gli SDK o il. AWS AWS CLI Per informazioni sui prezzi relativi a capacità con provisioning, consulta Prezzi di Amazon S3.

Frequenze delle richieste di ripristino S3 Glacier

Quando si avviano richieste di ripristino per oggetti archiviati nella classe di archiviazione S3 Glacier Flexible Retrieval (Recupero flessibile S3 Glacier) o S3 Glacier Deep Archive (Archiviazione profonda S3 Glacier), viene applicata una quota di richieste di recupero per Account AWS. S3 Glacier supporta richieste di ripristino a una velocità massima di 1.000 transazioni al secondo. Se questa velocità viene superata, le richieste altrimenti valide vengono sottoposte alla limitazione della larghezza di banda della rete o rifiutate e Amazon S3 restituisce un errore `ThrottlingException`.

Facoltativamente, puoi anche utilizzare Operazioni in batch S3 per recuperare un gran numero di oggetti archiviati nella classe S3 Glacier Flexible Retrieval (Recupero flessibile S3 Glacier) o S3 Glacier Deep Archive (Archiviazione profonda S3 Glacier) con un'unica richiesta. Per ulteriori informazioni, consulta Esecuzione di operazioni in batch su larga scala su oggetti Amazon S3.

Ripristino di un oggetto archiviato

Gli oggetti Amazon S3 nelle classi o nei livelli di archiviazione seguenti sono archiviati e non sono accessibili in tempo reale:

- Classe di archiviazione S3 Glacier Flexible Retrieval (Recupero flessibile S3 Glacier)
- Classe di archiviazione S3 Glacier Deep Archive (Archiviazione profonda S3 Glacier)
- Livello S3 Intelligent-Tiering Archive Access (Accesso archiviazione Piano intelligente S3)
- Livello Deep Archive Access di Piano intelligente Amazon S3

Gli oggetti Amazon S3 memorizzati nelle classi di archiviazione S3 Glacier Flexible Retrieval S3 Glacier Deep Archive non sono immediatamente accessibili. Per accedere a un oggetto in queste classi di archiviazione, è necessario ripristinare una copia temporanea dell'oggetto nel relativo bucket S3 per una durata specificata (numero di giorni). Se vuoi ottenere una copia permanente dell'oggetto, ripristina l'oggetto e creane quindi una copia nel bucket Amazon S3. L'operazione di copia degli oggetti ripristinati non è supportata nella console Amazon S3. Per questo tipo di operazione di copia,

usa AWS Command Line Interface (AWS CLI), gli SDK o l'API REST. AWS A meno che non si crei una copia, l'oggetto verrà comunque archiviato nelle classi di archiviazione S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive. Per informazioni sull'utilizzo di queste classi di archiviazione, consulta Classi di archiviazione per oggetti a cui si accede raramente.

Per accedere agli oggetti nei livelli Accesso archivio e Accesso archivio approfondito di S3 Intelligent-Tiering, è necessario avviare una richiesta di ripristino e attendere che l'oggetto venga spostato nel livello Frequent Access. Quando esegui il ripristino dai livelli Accesso di archiviazione o di archiviazione profonda, l'oggetto passa nuovamente al livello Accesso frequente. Per informazioni sull'utilizzo di queste classi di archiviazione, consulta Classe di storage per ottimizzare automaticamente i dati con modelli di accesso variabili o sconosciuti.

Per informazioni generali sugli oggetti archiviati, consulta Utilizzo di oggetti archiviati.

> **Note**
>
> - Quando ripristini un oggetto archiviato dalle classi di storage S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, paghi sia per l'oggetto archiviato che per la copia ripristinata temporaneamente.
> - Quando ripristini un oggetto da S3 Intelligent-Tiering, non sono previsti costi di recupero per i recuperi Standard o Bulk.
> - Le richieste di ripristino successive richiamate su oggetti archiviati che sono già stati ripristinati vengono fatturate come richieste. `GET` Per informazioni sui prezzi, consulta Prezzi di Amazon S3.

Ripristino di un oggetto archiviato

Puoi ripristinare un oggetto archiviato utilizzando la console Amazon S3, l'API REST di Amazon S3, gli SDK, AWS AWS CLI() o S3 Batch AWS Command Line Interface Operations.

Utilizzo della console S3

Ripristino di oggetti mediante la console Amazon S3

Usa la seguente procedura per ripristinare un oggetto che è stato archiviato nelle classi di archiviazione S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive o nei livelli di archiviazione Accesso archivio e Accesso archivio approfondito di S3 Intelligent-Tiering.

Per ripristinare un oggetto archiviato

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckets (Bucket) scegliere il nome del bucket che contiene gli oggetti che si desidera ripristinare.
4. Nell'elenco Objects (Oggetti) selezionare l'oggetto o gli oggetti che si desidera ripristinare, scegliere Actions (Operazioni), quindi selezionare Initiate restore (Avvia ripristino).
5. Se esegui il ripristino da S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, immetti il numero di giorni in cui desideri che i dati archiviati siano accessibili nella casella Numero di gierni in cui la copia ripristinata è disponibile.
6. Per Opzioni di recupero, effettua una delle seguenti operazioni:

   - Sceglie Recupero Bulk oppure Recupero Standard, quindi seleziona Ripristina.
   - Scegli Expedited retrieval (Recupero rapido) (disponibile solo per S3 Glacier Flexible Retrieval o S3 Intelligent-Tiering Archive Access). Se stai ripristinando un oggetto in S3 Glacier Flexible Retrieval, puoi scegliere se acquistare capacità assegnata per il recupero Expedited. Se desideri acquistare capacità assegnata, procedi alla fase successiva. Altrimenti, scegli Avvia il ripristino.

   > **Note**
   >
   > Gli oggetti dei livelli Accesso archivio e Accesso archivio approfondito di S3 Intelligent-Tiering vengono ripristinati automaticamente al livello Frequent Access.

7. (Facoltativo) Se stai ripristinando un oggetto in S3 Glacier Flexible Retrieval e scegli Recupero expedited puoi scegliere se acquistare capacità assegnata. La capacità assegnata è disponibile solo per gli oggetti in S3 Glacier Flexible Retrieval. Se disponi già di capacità assegnata, scegli Ripristina per avviare un ripristino mediante capacità assegnata.

   Se disponi di capacità assegnata, tutti i recuperi Expedited vengono eseguiti mediante tale capacità assegnata. Per ulteriori informazioni, consulta Capacità con provisioning.

   - Se non disponi di capacità assegnata e non desideri acquistarla, scegli Ripristina.

- Se non disponi di capacità assegnata, ma desideri acquistare unità di capacità assegnata (PCU), scegli Acquisto di PCU. Nella finestra di dialogo Acquisto di PCU, scegli quante PCU vuoi acquistare, conferma l'acquisto e poi scegli Acquisto di PCU. Quando ricevi il messaggio Acquisto riuscito, scegli Ripristina per avviare un recupero mediante capacità assegnata.

Usando il AWS CLI

Ripristino di oggetti da S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive

L'esempio seguente utilizza il comando `restore-object` per ripristinare l'oggetto *`dir1/example.obj`* nel bucket *`example-s3-bucket`* per 25 giorni.

```
aws s3api restore-object --bucket example-s3-bucket --key dir1/example.obj --restore-request '{"Days":25,"GlacierJobParameters":{"Tier":"Standard"}}'
```

Se la sintassi JSON utilizzata nell'esempio genera un errore su un client Windows, sostituire la richiesta di ripristino con la seguente sintassi:

```
--restore-request Days=25,GlacierJobParameters={"Tier"="Standard"}
```

Ripristino di oggetti da Accesso archivio e Accesso archivio approfondito di S3 Intelligent-Tiering

L'esempio seguente utilizza il comando `restore-object` per ripristinare l'oggetto *`dir1/example.obj`* nel bucket *`example-s3-bucket`* nel livello Frequent Access.

```
aws s3api restore-object --bucket example-s3-bucket --key dir1/example.obj --restore-request '{}'
```

> **Note**
>
> A differenza delle classi di archiviazione di S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive, le richieste di ripristino per gli oggetti S3 Intelligent-Tiering non accettano il valore `Days`.

Monitoraggio dello stato del ripristino

Per monitorare lo stato della richiesta `restore-object`, usa il seguente comando `head-object`:

```
aws s3api head-object --bucket example-s3-bucket --key dir1/example.obj
```

Per ulteriori informazioni, consulta la sezione restore-object nella Documentazione di riferimento della AWS CLI .

Utilizzo di REST API

Amazon S3 fornisce un'operazione API che consente di avviare il ripristino di un oggetto archiviato. Per ulteriori informazioni, consulta RestoreObject in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Utilizzo degli AWS SDK

Per esempi su come ripristinare oggetti archiviati in S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive con gli SDK, consulta. AWS Utilizzo `RestoreObject` con un AWS SDK o una CLI

Utilizzo delle operazioni in batch S3

Per ripristinare più di un oggetto archiviato con una sola richiesta, puoi utilizzare le operazioni in batch S3. Fornisci alle operazioni in batch S3 un elenco di oggetti su cui operare. Le operazioni in batch S3 richiamano la rispettiva API per eseguire l'operazione specificata. Un solo processo di operazioni in batch può eseguire l'operazione specificata su miliardi di oggetti contenenti esabyte di dati.

Per creare un processo di operazioni in batch, è necessario disporre di un manifesto che contenga solo gli oggetti che si desidera ripristinare. Puoi creare un manifesto utilizzando inventario S3 oppure puoi fornire un file CSV con le informazioni necessarie. Per ulteriori informazioni, consulta the section called “Specifica di un manifest”.

Prima di creare ed eseguire i processi delle operazioni in batch S3, devi concedere le autorizzazioni ad Amazon S3 per eseguire tali operazioni per tuo conto. Per le autorizzazioni richieste, consulta the section called “Concessione di autorizzazioni”.

> **Note**
>
> I processi delle operazioni in batch possono funzionare su oggetti di classe di archiviazione S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive o su oggetti di livello di archiviazione Accesso archivio e Accesso archivio approfondito di S3 Intelligent-Tiering. Le operazioni in batch non possono operare su entrambi i tipi di oggetti archiviati nello stesso processo. Per ripristinare oggetti di entrambi i tipi, devi creare processi Batch Operations separati.

Per ulteriori informazioni sull'utilizzo delle operazioni in batch per la replica di oggetti esistenti, consulta the section called “Ripristino di oggetti”.

Creazione di un processo di operazioni in batch S3 Initiate Restore Object

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Operazioni in batch.
3. Scegli Crea processo.
4. Per Regione AWS, scegli la regione in cui creare il processo.
5. In Formato manifest scegli il tipo di oggetto manifesto da usare.
   - Se scegli Report di inventario S3, inserisci il percorso dell'oggetto `manifest.json` generato da Amazon S3 come parte del report di inventario in formato CSV. Se desideri utilizzare una versione del manifesto diversa da quella più recente, immetti l'ID della versione dell'oggetto `manifest.json`.
   - Se si sceglie CSV, immettere il percorso di un oggetto manifest in formato CSV. L'oggetto manifest deve avere il formato descritto nella console. Se desideri utilizzare una versione diversa da quella più recente, puoi includere facoltativamente l'ID della versione dell'oggetto manifesto.
6. Seleziona Successivo.
7. Nella sezione Operazione, scegli Ripristina.
8. Nella sezione Ripristina, per Ripristina origine, scegli Glacier Flexible Retrieval o Glacier Deep Archive oppure il livello Accesso archivio e Accesso archivio approfondito di Intelligent-Tiering.

   Se hai scelto Glacier Flexible Retrieval o Glacier Deep Archive, inserisci un numero per Numero di giorni in cui la copia ripristinata è disponibile.

   Per Livello di recupero, scegli il livello che desideri utilizzare.
9. Seleziona Successivo.
10. Nella pagina Configura opzioni aggiuntive, compila le seguenti sezioni:
    - Nella sezione Altre opzioni, fornisci una descrizione del processo e specifica un numero di priorità per il processo. I numeri più alti indicano una priorità più alta. Per ulteriori informazioni, consulta the section called “Assegnazione della priorità dei processi”.

- Nella sezione Report di completamento, seleziona se le operazioni in batch devono creare un report di completamento. Per ulteriori informazioni sui report di completamento, consulta the section called “Rapporti di completamento”.
- Nella sezione Autorizzazioni, devi concedere ad Amazon S3 le autorizzazioni per eseguire operazioni in batch per tuo conto. Per le autorizzazioni richieste, consulta the section called “Concessione di autorizzazioni”.
- (Facoltativo) Nella sezione Tag dell'attività, aggiungi tag nelle coppie chiave-valore. Per ulteriori informazioni, consulta the section called “Utilizzo dei tag”.

Quando hai terminato, seleziona Successivo.

11. Nella pagina Review (Rivedi), verificare le impostazioni. Se è necessario apportare modifiche, scegliere Previous (Precedente). In caso contrario, scegli Crea processo.

Per ulteriori informazioni sulle operazioni in batch, consulta Ripristino di oggetti con operazioni in batch e Creazione di un processo di operazioni in batch S3.

Verifica dello stato del ripristino e della data di scadenza

Puoi controllare lo stato di una richiesta di ripristino o la data di scadenza utilizzando la console Amazon S3, Amazon S3 Event Notifications o AWS CLI l'API REST di Amazon S3.

> **Note**
>
> Gli oggetti ripristinati dalle classi di storage S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive vengono archiviati solo per il numero di giorni specificato. Le seguenti procedure restituiscono la data di scadenza di queste copie.
> Gli oggetti ripristinati dai livelli di storage S3 Intelligent-Tiering Archive Access e Deep Archive Access non hanno date di scadenza e vengono invece riportati al livello Frequent Access.

Utilizzo della console S3

Verifica dello stato di ripristino e della data di scadenza di un oggetto nella console Amazon S3

1. Apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket scegli il nome del bucket che contiene gli oggetti che desideri ripristinare.

4. Nell'elenco Oggetti, seleziona l'oggetto che stai ripristinando. Viene visualizzata la pagina dei dettagli dell'oggetto.

   - Se il ripristino non è terminato, nella parte superiore della pagina, viene visualizzata una sezione che indica Ripristino in corso.
   - Se il ripristino è terminato, nella parte superiore della pagina, viene visualizzata una sezione che indica Ripristino completo. Se esegui il ripristino da S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, questa sezione indica anche la Data di scadenza del ripristino. In questa data Amazon S3 rimuoverà la copia ripristinata dell'oggetto archiviato.

Utilizzo delle notifiche di eventi di Amazon S3

Puoi ricevere una notifica del completamento del ripristino degli oggetti utilizzando l'`s3:ObjectRestore:Completed`azione con la funzionalità Amazon S3 Event Notifications. Per ulteriori informazioni sull'attivazione delle notifiche di eventi, consulta Abilitare le notifiche utilizzando Amazon SQS, Amazon SNS e. AWS Lambda Per ulteriori informazioni sui vari tipi di `ObjectRestore` eventi, consulta. the section called “Tipi di eventi supportati per SQS, SNS e Lambda”

Usando il AWS CLI

Controlla lo stato di ripristino e la data di scadenza di un oggetto con AWS CLI

L'esempio seguente utilizza il comando `head-object` per visualizzare i metadati dell'oggetto *`dir1/example.obj`* nel bucket *`example-s3-bucket`*. Quando esegui questo comando su un oggetto in fase di ripristino, Amazon S3 indica se il ripristino è in corso e (se applicabile) la data di scadenza.

```
aws s3api head-object --bucket example-s3-bucket --key dir1/example.obj
```

Output previsto (ripristino in corso):

```
{
    "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"
}
```

Output previsto (ripristino terminato):

```
{
    "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"
}
```

Per ulteriori informazioni in meritohead-object, vedere head-objectnel AWS CLI Command Reference.

Utilizzo di REST API

Amazon S3 fornisce un'operazione API per recuperare i metadati degli oggetti. Per verificare lo stato di ripristino e la data di scadenza di un oggetto archiviato utilizzando la REST API, consulta HeadObject in Amazon Simple Storage Service API Reference.

Aggiornamento della velocità di un ripristino in corso

Puoi aggiornare la velocità di un ripristino mentre il ripristino è in corso.

Per aggiornare un ripristino in corso a un livello più veloce

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckets (Bucket) scegliere il nome del bucket che contiene gli oggetti che si desidera ripristinare.
4. Nell'elenco Oggetti, seleziona l'oggetto che stai ripristinando. Viene visualizzata la pagina dei dettagli dell'oggetto. Nella pagina dei dettagli dell'oggetto, scegli Aggiorna livello di recupero. Per informazioni su come verificare lo stato del ripristino di un oggetto, consulta Verifica dello stato del ripristino e della data di scadenza.

5. Seleziona il livello che desideri aggiornare, quindi seleziona Avvia il ripristino.

Utilizzo del blocco oggetti S3

Object Lock S3 può impedire che gli oggetti Amazon S3 vengano eliminati o sovrascritti per un determinato periodo di tempo o in modo indefinito. Object Lock utilizza un modello write-once-read-many(WORM) per archiviare oggetti. È possibile utilizzare Object Lock per soddisfare i requisiti normativi che richiedono lo storage WORM o per aggiungere un altro livello di protezione contro le modifiche o l'eliminazione degli oggetti.

> **Note**
>
> Object Lock S3 è stato valutato da Cohasset Associates per l'utilizzo in ambienti soggetti alle normative SEC 17a-4, CTCC e FINRA. Per ulteriori informazioni su come Object Lock fa riferimento a queste normative, consulta Cohasset Associates Compliance Assessment.

Il blocco degli oggetti offre due modi per gestire la conservazione degli oggetti: i periodi di conservazione e i blocchi a fini giudiziari. La versione di un oggetto può avere un periodo di conservazione, un blocco a fini legali o entrambi.

- Periodo di conservazione: un periodo di conservazione specifica un determinato intervallo di tempo durante il quale un oggetto rimane bloccato. È possibile impostare un periodo di conservazione univoco per singoli oggetti. Inoltre, puoi impostare un periodo di conservazione predefinito su un bucket S3. Puoi anche limitare i periodi di conservazione minimi e massimi consentiti utilizzando la chiave `s3:object-lock-remaining-retention-days` condition nella policy del bucket. Ciò consente di stabilire un intervallo di periodi di conservazione e di limitare i periodi di conservazione che possono essere più brevi o più lunghi di questo intervallo.
- Blocco a fini legali: un blocco a fini legali offre la stessa protezione di un periodo di conservazione ma non presenta una data di scadenza. pertanto rimane invariato fino a quando non viene rimosso esplicitamente. Le conservazioni legali sono indipendenti dai periodi di conservazione e vengono applicate alle versioni dei singoli oggetti.

Object Lock funziona solo nei bucket con il controllo delle versioni S3 abilitato. Quando si blocca una versione di un oggetto, Amazon S3 archivia le informazioni di blocco nei metadati per quella versione di oggetto. L'inserimento di un periodo di conservazione o di un blocco a fini legali in un oggetto

protegge solo la versione specificata nella richiesta. I periodi di conservazione e i blocchi legali non impediscono la creazione di nuove versioni dell'oggetto o l'eliminazione dei marcatori da aggiungere sopra l'oggetto. Per informazioni sulla funzione Controllo delle versioni S3, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Se si inserisce un oggetto in un bucket che contiene già un oggetto protetto esistente con lo stesso nome della chiave dell'oggetto, Amazon S3 crea una nuova versione dell'oggetto. La versione protetta esistente dell'oggetto rimane bloccata in base alla rispettiva configurazione di conservazione.

Come funziona il blocco oggetti S3

Argomenti

Periodi di conservazione

Un periodo di conservazione protegge una versione di un oggetto per un determinato intervallo di tempo. Quando imposti un periodo di conservazione per una versione di un oggetto, Amazon S3 archivia un timestamp nei metadati della versione dell'oggetto per indicare la scadenza del periodo di conservazione. Allo scadere del periodo di conservazione, la versione dell'oggetto può essere sovrascritta o eliminata.

È possibile inserire un periodo di conservazione in modo esplicito sulla versione di un singolo oggetto o sulle proprietà di un bucket in modo che si applichi automaticamente a tutti gli oggetti nel bucket. Quando applichi un periodo di conservazione a una versione di un oggetto in modo esplicito, specifichi una Data di fine conservazione per tale versione. Amazon S3 archivia questa data nei metadati della versione dell'oggetto.

È anche possibile impostare un periodo di conservazione nelle proprietà di un bucket. Quando si imposta un periodo di conservazione su un bucket, si specifica una durata, in giorni o in anni, per la protezione di qualsiasi versione dell'oggetto inserita nel bucket. Quando si inserisce un oggetto nel bucket, Amazon S3 calcola una Data di fine conservazione per la versione dell'oggetto aggiungendo la durata specificata al timestamp di creazione della versione dell'oggetto. La versione dell'oggetto

viene quindi protetta esattamente come se fosse stato impostato un singolo blocco con tale periodo di conservazione sulla versione dell'oggetto.

> **Note**
>
> Quando esegui il PUT di una versione dell'oggetto che dispone di una modalità e un periodo di conservazione individuali espliciti in un bucket, le impostazioni Object Lock individuali della versione dell'oggetto hanno la precedenza su qualsiasi impostazione di conservazione delle proprietà del bucket.

Come tutte le altre impostazioni del blocco degli oggetti, i periodi di conservazione si applicano alle singole versioni degli oggetti. Versioni diverse di un singolo oggetto possono avere modalità e periodi di conservazione diversi.

Ad esempio, supponi di avere un oggetto che è a 15 giorni di un periodo di conservazione di 30 giorni e applichi il comando PUT a un oggetto in Amazon S3 con lo stesso nome e un periodo di conservazione di 60 giorni. In questo caso la richiesta PUT va a buon fine e Amazon S3 crea una nuova versione dell'oggetto con un periodo di conservazione di 60 giorni. Per la versione precedente rimane impostato il periodo di conservazione originale e tale versione può quindi essere eliminata in 15 giorni.

Dopo aver applicato un'impostazione di conservazione a una versione dell'oggetto, è possibile estendere il periodo di conservazione. A tale scopo, invia una nuova richiesta Object Lock per la versione dell'oggetto con una Data di fine conservazione posteriore rispetto a quella attualmente configurata per la versione dell'oggetto. Amazon S3 sostituisce il periodo di conservazione esistente con il nuovo periodo più lungo. Qualsiasi utente con autorizzazioni per impostare un periodo di conservazione per un oggetto può estendere il periodo di conservazione per una versione di un oggetto. Per impostare un periodo di conservazione, è necessaria l'autorizzazione `s3:PutObjectRetention`.

Quando si imposta un periodo di conservazione su un oggetto o bucket S3, è necessario selezionare una delle due modalità di conservazione: conformità o governance.

Modalità di conservazione

Object Lock S3 fornisce due modalità di conservazione che applicano livelli di protezione diversi agli oggetti:

- Modalità Conformità
- Modalità Governance

In modalità conformità, una versione protetta di un oggetto non può essere sovrascritta o eliminata da alcun utente, incluso l'utente root in Account AWS. Quando un oggetto è bloccato in modalità conformità, la relativa modalità di conservazione non può essere modificata e il periodo di conservazione non può essere abbreviato. La modalità conformità garantisce che una versione di un oggetto non possa essere sovrascritta o eliminata per tutta la durata del periodo di conservazione.

> **Note**
>
> L'unico modo per eliminare un oggetto in modalità di conformità prima della scadenza della data di conservazione è eliminare l'oggetto associato. Account AWS

Nella modalità Governance, gli utenti non possono sovrascrivere o eliminare una versione di un oggetto, né modificare le relative impostazioni di blocco, a meno che non dispongano di autorizzazioni speciali. La modalità governance permette di impedire alla maggior parte degli utenti di eliminare gli oggetti ma, allo stesso tempo, concede ad alcuni utenti l'autorizzazione per modificare le impostazioni di conservazione o per eliminare gli oggetti, se necessario. Puoi usare la modalità governance anche per testare le impostazioni del periodo di conservazione prima di creare un periodo di conservazione in modalità di conformità.

Per sovrascrivere o rimuovere le impostazioni di conservazione in modalità governance, occorre disporre dell'autorizzazione `s3:BypassGovernanceRetention` e includere in modo esplicito `x-amz-bypass-governance-retention:true` come un'intestazione della richiesta in qualsiasi richiesta che richieda la sostituzione della modalità governance.

> **Note**
>
> Per impostazione predefinita, la console Amazon S3 include l'intestazione `x-amz-bypass-governance-retention:true`. Se si prova a eliminare oggetti protetti dalla modalità governance e che dispongono dell'autorizzazione `s3:BypassGovernanceRetention`, l'operazione andrà a buon fine.

Blocchi a fini giudiziari

Con Object Lock è possibile inserire anche un blocco a fini legali nella versione di un oggetto. Analogamente a un periodo di conservazione, un blocco a fini giudiziari impedisce che una versione di un oggetto venga sovrascritta o eliminata. Tuttavia, un blocco a fini legali non dispone di un periodo di tempo fisso associato e rimane valido fino a quando non viene rimosso. I blocchi a fini giudiziari possono essere applicati e rimossi liberamente da qualsiasi utente con l'autorizzazione `s3:PutObjectLegalHold`.

I blocchi a fini giudiziari sono indipendenti dai periodi di conservazione. L'applicazione di un blocco a fini giudiziari a una versione di un oggetto non influisce sulla modalità di conservazione o sul periodo di conservazione per tale versione dell'oggetto.

Ad esempio, supponi di inserire un blocco a fini legali nella versione di un oggetto mentre la versione dell'oggetto è protetta da un periodo di conservazione. Se il periodo di conservazione scade, l'oggetto non perde la protezione WORM. Il blocco a fini legali continua a proteggere l'oggetto fino a quando non viene rimosso in modo esplicito da un utente autorizzato. Analogamente, se rimuovi un blocco a fini giudiziari da una versione di un oggetto per la quale è impostato un periodo di conservazione, la versione dell'oggetto rimane protetta fino alla scadenza del periodo di conservazione.

Le migliori pratiche per l'utilizzo di S3 Object Lock

Prendi in considerazione l'utilizzo della modalità Governance se desideri proteggere gli oggetti dall'eliminazione da parte della maggior parte degli utenti durante un periodo di conservazione predefinito, ma allo stesso tempo desideri che alcuni utenti con autorizzazioni speciali abbiano la flessibilità necessaria per modificare le impostazioni di conservazione o eliminare gli oggetti.

Prendi in considerazione l'utilizzo della modalità Compliance se non desideri che nessun utente, incluso l'utente root del tuo Account AWS, sia in grado di eliminare gli oggetti durante un periodo di conservazione predefinito. È possibile utilizzare questa modalità nel caso in cui sia necessario archiviare dati conformi.

Puoi usare Legal Hold quando non sei sicuro per quanto tempo desideri che i tuoi oggetti rimangano immutabili. Ciò potrebbe essere dovuto al fatto che è imminente un controllo esterno dei dati e desideri mantenere gli oggetti immutabili fino al completamento del controllo. In alternativa, potresti avere un progetto in corso che utilizza un set di dati che desideri mantenere immutabile fino al completamento del progetto.

Autorizzazioni richieste

Le operazioni del blocco degli oggetti richiedono autorizzazioni specifiche. A seconda dell'operazione esatta che si sta tentando di eseguire, potrebbe essere necessaria una delle seguenti autorizzazioni:

- `s3:BypassGovernanceRetention`
- `s3:GetBucketObjectLockConfiguration`
- `s3:GetObjectLegalHold`
- `s3:GetObjectRetention`
- `s3:PutBucketObjectLockConfiguration`
- `s3:PutObjectLegalHold`
- `s3:PutObjectRetention`

Per un elenco completo delle autorizzazioni di Amazon S3 con descrizioni, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.

Per informazioni sull'utilizzo delle condizioni con le autorizzazioni, consulta Esempi di policy Bucket che utilizzano chiavi condizionali.

Considerazioni su Object Lock

Amazon S3 Object Lock può impedire che gli oggetti vengano eliminati o sovrascritti per un determinato periodo di tempo o in modo indefinito.

Puoi utilizzare la console Amazon S3, AWS Command Line Interface (AWS CLI), gli AWS SDK o l'API REST di Amazon S3 per visualizzare o impostare le informazioni di Object Lock. Per informazioni generali sulle funzionalità S3 Object Lock, consulta Utilizzo del blocco oggetti S3.

Important

- Dopo aver abilitato Object Lock su un bucket, non è possibile disabilitare Object Lock o sospendere il controllo delle versioni per tale bucket.
- I bucket S3 con Object Lock non possono essere utilizzati come bucket di destinazione per i log di accesso al server. Per ulteriori informazioni, consulta the section called “Registrazione dell'accesso al server”.

Argomenti

Autorizzazioni per la visualizzazione di informazioni di blocco

È possibile visualizzare in modo programmatico lo stato Object Lock della versione di un oggetto Amazon S3 mediante le operazioni HeadObject o GetObject. Entrambe le operazioni restituiscono la modalità di conservazione, la data di fine conservazione e lo stato del blocco a fini legali per la versione dell'oggetto specificata. Inoltre, puoi visualizzare lo stato di Object Lock per più oggetti nel tuo bucket S3 utilizzando S3 Inventory.

Per visualizzare la modalità e il periodo di conservazione della versione di un oggetto, è necessaria l'autorizzazione `s3:GetObjectRetention`. Per visualizzare lo stato di blocco per vincoli di legge di un oggetto, è necessaria l'autorizzazione `s3:GetObjectLegalHold`. Per visualizzare la configurazione di conservazione predefinita di un bucket, occorre disporre dell'autorizzazione `s3:GetBucketObjectLockConfiguration`. Se si esegue una richiesta per una configurazione Object Lock su un bucket che non dispone di S3 Object Lock abilitato, Amazon S3 restituisce un errore.

Bypassare la modalità Governance

Se si dispone dell'autorizzazione `s3:BypassGovernanceRetention`, è possibile eseguire operazioni su versioni degli oggetti bloccate nella modalità governance come se non fossero protette. Queste operazioni includono l'eliminazione di una versione dell'oggetto, la riduzione del periodo di conservazione o la rimozione del periodo di conservazione di Object Lock tramite l'inserimento di una nuova richiesta `PutObjectRetention` con parametri vuoti.

Per bypassare la modalità Governance, è necessario indicare esplicitamente nella richiesta che si desidera bypassare la modalità Governance. A tale scopo, includi l'`x-amz-bypass-governance-retention:true`intestazione nella richiesta di operazione `PutObjectRetention` API o utilizza il parametro equivalente con le richieste effettuate tramite o SDK. AWS CLI AWS La console S3 applica automaticamente questa intestazione alle richieste effettuate tramite la console S3 se si dispone dell'autorizzazione `s3:BypassGovernanceRetention`.

> **Note**
>
> Bypassare la modalità Governance non modifica lo stato dei vincoli di legge della versione di un oggetto. Se nella versione di un oggetto è abilitato un blocco a fini legali, questo rimane in vigore e impedisce richieste di sovrascrittura o eliminazione della versione dell'oggetto.

Utilizzo di Object Lock con la replica S3

È possibile utilizzare Object Lock con la replica S3 per abilitare la copia asincrona e automatica di oggetti bloccati e dei relativi metadati di conservazione tra i bucket S3. Ciò significa che per gli oggetti replicati, Amazon S3 utilizza la configurazione di blocco degli oggetti del bucket di origine. In altre parole, se il bucket di origine ha Object Lock abilitato, anche i bucket di destinazione devono avere Object Lock abilitato. Se un oggetto viene caricato direttamente nel bucket di destinazione (al di fuori di S3 Replication), richiede l'Object Lock impostato sul bucket di destinazione. Quando si utilizza la replica, gli oggetti in un bucket di origine vengono replicati in uno o più bucket di destinazione.

Per configurare la replica su un bucket con Object Lock abilitato, puoi utilizzare la console S3, l'API REST di Amazon AWS CLI S3 o gli SDK. AWS

> **Note**
>
> Per utilizzare Object Lock con la replica, devi concedere due autorizzazioni aggiuntive sul bucket S3 di origine nel ruolo AWS Identity and Access Management (IAM) che usi per configurare la replica. Le due nuove autorizzazioni aggiuntive sono `s3:GetObjectRetention` e `s3:GetObjectLegalHold`. Se il ruolo dispone di un'istruzione di autorizzazione `s3:Get*`, tale istruzione soddisfa il requisito. Per ulteriori informazioni, consulta Impostazione delle autorizzazioni per la replica in tempo reale.
> Per informazioni generali sulla replica S3, consulta Panoramica sulla replica degli oggetti.
> Per esempi di configurazione della replica S3, consulta Esempi di configurazione della replica in tempo reale.

Utilizzo di Object Lock con Inventario Amazon S3

È possibile configurare Inventario Amazon S3 per creare elenchi degli oggetti in un bucket S3 in base a una pianificazione definita. È possibile configurare Inventario Amazon S3 per includere i seguenti metadati Object Lock per gli oggetti:

- La data di fine conservazione
- La modalità di conservazione
- Lo stato di blocco a fini legali

Per ulteriori informazioni, consulta Amazon S3 Inventory.

Gestione delle policy del ciclo di vita di S3 con Object Lock

Le configurazioni di gestione del ciclo di vita di un oggetto continuano a funzionare normalmente sugli oggetti protetti, compresa l'applicazione del contrassegno di eliminazione. Tuttavia, una versione bloccata di un oggetto non può essere eliminata da una politica di scadenza di S3 Lifecycle. Object Lock viene mantenuto indipendentemente dalla classe di storage in cui risiede l'oggetto e durante le transizioni del ciclo di vita di S3 tra le classi di storage.

Per ulteriori informazioni sulla gestione della configurazione del ciclo di vita di un oggetto, consulta Gestione del ciclo di vita dello storage.

Gestione dei marker di eliminazione con Object Lock

Anche se non è possibile eliminare una versione protetta di un oggetto, puoi comunque creare un contrassegno di eliminazione per tale oggetto. L'inserimento di un contrassegno di eliminazione su un oggetto non elimina l'oggetto né alcuna sua versione. Tuttavia, fa sì che Amazon S3 si comporti per molti versi come se l'oggetto fosse stato eliminato. Per ulteriori informazioni, consulta Utilizzo dei contrassegni di eliminazione.

> **Note**
>
> I contrassegni di eliminazione non sono protetti contro i WORM, indipendentemente dal periodo di conservazione o dal blocco per vincoli di legge dell'oggetto a cui si riferiscono.

Utilizzo di S3 Storage Lens con Object Lock

Per visualizzare i parametri relativi ai byte di archiviazione abilitati per il blocco e il conteggio degli oggetti, puoi utilizzare Amazon S3 Storage Lens. S3 Storage Lens è una funzionalità di analisi dell'archiviazione su cloud che puoi utilizzare per avere una panoramica completa a livello di organizzazione sull'utilizzo e sulle attività relative all'archiviazione di oggetti.

Per ulteriori informazioni, consulta Utilizzo di S3 Storage Lens per proteggere i tuoi dati.

Per un elenco completo di parametri, consulta Glossario dei parametri di Amazon S3 Storage Lens.

Caricamento di oggetti in un bucket abilitato a Object Lock

L'`Content-MD5`intestazione è necessaria per qualsiasi richiesta di caricamento di un oggetto con un periodo di conservazione configurato utilizzando Object Lock. Il digest MD5 è un modo per verificare l'integrità dell'oggetto dopo averlo caricato in un bucket. Dopo aver caricato l'oggetto, Amazon S3 calcola il digest MD5 dell'oggetto e lo confronta con il valore fornito. La richiesta ha esito positivo solo se i due digest corrispondono. La console S3 aggiunge automaticamente questa intestazione, tuttavia è necessario specificare questa intestazione quando si utilizza l'API. PutObject

Per ulteriori informazioni, consulta Utilizzo di Content-MD5 durante il caricamento di oggetti.

Configurare eventi e notifiche

Puoi utilizzare Amazon S3 Event Notifications per tenere traccia degli accessi e delle modifiche alle configurazioni e ai dati di Object Lock utilizzando. AWS CloudTrail Per informazioni su CloudTrail, consulta What is? AWS CloudTrail nella Guida AWS CloudTrail per l'utente.

Puoi anche utilizzare Amazon CloudWatch per generare avvisi basati su questi dati. Per informazioni su CloudWatch, consulta What is Amazon CloudWatch? nella Amazon CloudWatch User Guide.

Impostazione di limiti su periodi di conservazione con una policy di bucket

Puoi impostare periodi di conservazione minimo e massimo per un bucket mediante una policy di bucket. Il periodo massimo di conservazione è 100 anni.

L'esempio seguente mostra una policy di bucket che utilizza la chiave di condizione `s3:object-lock-remaining-retention-days` per impostare un periodo di conservazione massimo di 10 giorni.

```
{
```

```
    "Version": "2012-10-17",
    "Id": "SetRetentionLimits",
    "Statement": [
        {
            "Sid": "SetRetentionPeriod",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "s3:PutObjectRetention"
            ],
            "Resource": "arn:aws:s3:::example-s3-bucket1/*",
            "Condition": {
                "NumericGreaterThan": {
                    "s3:object-lock-remaining-retention-days": "10"
                }
            }
        }
    ]
}
```

> **Note**
>
> Se il bucket è quello di destinazione per una configurazione di replica, puoi impostare i periodi di conservazione minimo e massimo per le repliche di oggetti creati mediante la replica. A questo scopo, occorre consentire l'operazione `s3:ReplicateObject` nella policy di bucket. Per ulteriori informazioni sulle autorizzazioni di replica, consulta the section called “Impostazione delle autorizzazioni”.

Per ulteriori informazioni sulle policy di bucket, consulta gli argomenti indicati di seguito:

- Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference
- Operazioni sugli oggetti
- Esempi di policy Bucket che utilizzano chiavi condizionali

Configurazione di S3 Object Lock

Con Amazon S3 Object Lock, puoi archiviare oggetti in Amazon S3 utilizzando write-once-read-manyun modello (WORM). Puoi utilizzarlo per impedire che un oggetto venga eliminato o sovrascritto

per un periodo di tempo fisso o indefinito. Per informazioni generali sulle funzionalità Object Lock, consulta Utilizzo del blocco oggetti S3.

Prima di bloccare eventuali oggetti, devi abilitare il controllo delle versioni S3 e Object Lock su un bucket. In seguito, puoi impostare un periodo di conservazione, un blocco a fini legali o entrambi.

Per utilizzare Object Lock, devi disporre di determinate autorizzazioni. Per un elenco delle autorizzazioni correlate a varie operazioni Object Lock, consulta the section called “Autorizzazioni richieste”.

⚠ Important

- Dopo aver abilitato Object Lock su un bucket, non è possibile disabilitare Object Lock o sospendere il controllo delle versioni per tale bucket.
- I bucket S3 con Object Lock non possono essere utilizzati come bucket di destinazione per i log di accesso al server. Per ulteriori informazioni, consulta the section called “Registrazione dell'accesso al server”.

Argomenti

Abilitazione di Object Lock durante la creazione di un nuovo bucket S3

Puoi abilitare Object Lock durante la creazione di un nuovo bucket S3 utilizzando la console Amazon S3 AWS Command Line Interface ,AWS CLI() AWS , gli SDK o l'API REST di Amazon S3.

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).

3. Scegliere Create bucket (Crea bucket).

   Viene visualizzata la pagina Create bucket (Crea bucket).

4. In Nome bucket, immettere il nome del bucket.

   > **Note**
   >
   > Una volta creato un bucket, non è possibile modificarne il nome. Per ulteriori informazioni sulla denominazione dei bucket, consulta Regole di denominazione dei bucket.

5. Per Regione, scegli Regione AWS dove vuoi che risieda il bucket.
6. In Proprietà dell'oggetto, scegli di disabilitare o abilitare le liste di controllo degli accessi (ACL) e controllare la proprietà degli oggetti caricati nel bucket.
7. In Impostazioni di blocco dell'accesso pubblico per questo bucket scegli le impostazioni di blocco dell'accesso pubblico che vuoi applicare al bucket.
8. In Controllo delle versioni per il bucket, scegli Abilitato.

   Object Lock funziona solo con bucket con versioni.

9. (Facoltativo) In Tags (Tag), puoi scegliere di aggiungere tag al bucket. I tag sono coppie chiave-valore utilizzate per classificare lo spazio di archiviazione e allocare i costi.
10. In Impostazioni avanzate, trova Object Lock e scegli Attiva.

    Devi confermare che l'attivazione di Object Lock consentirà in modo permanente il blocco degli oggetti in questo bucket.

11. Seleziona Crea bucket.

Usando il AWS CLI

L'esempio `create-bucket` seguente crea un nuovo bucket S3 denominato *`example-s3-bucket1`* con Object Lock abilitato:

```
aws s3api create-bucket --bucket example-s3-bucket1 --object-lock-enabled-for-bucket
```

Per ulteriori informazioni ed esempi, consulta create-bucket nel Riferimento ai comandi AWS CLI .

> **Note**
>
> È possibile eseguire AWS CLI comandi dalla console utilizzando AWS CloudShell. AWS CloudShell è una shell preautenticata basata su browser che è possibile avviare direttamente da. AWS Management ConsolePer ulteriori informazioni, consulta Cos'è? CloudShell nella Guida AWS CloudShell per l'utente.

Utilizzo di REST API

Puoi utilizzare la REST API per creare un nuovo bucket S3 con Object Lock abilitato. Per ulteriori informazioni, consulta CreateBucket in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Utilizzo degli AWS SDK

Per esempi su come abilitare Object Lock durante la creazione di un nuovo bucket S3 con gli AWS SDK, consulta. Utilizzo `CreateBucket` con un AWS SDK o una CLI

Per esempi su come ottenere la configurazione corrente di Object Lock con gli AWS SDK, consulta. Utilizzo `GetObjectLockConfiguration` con un AWS SDK o una CLI

Per uno scenario interattivo che illustra le diverse funzionalità di Object Lock utilizzando gli AWS SDK, consulta. Lavora con le funzionalità di blocco degli oggetti di Amazon S3 utilizzando un SDK AWS

Per informazioni generali sull'utilizzo di diversi AWS SDK, consulta. Sviluppo con Amazon S3 utilizzando gli SDK AWS

Abilitazione di Object Lock su un bucket S3 esistente

Puoi abilitare Object Lock per un bucket S3 esistente utilizzando la console Amazon S3, gli SDK o AWS CLI l'API AWS REST di Amazon S3.

Utilizzo della console S3

> **Note**
>
> Object Lock funziona solo con bucket con versioni.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket, scegli il nome del bucket per il quale desideri abilitare Object Lock.
4. Scegliere la scheda Properties (Proprietà).
5. In Proprietà, scorri verso il basso fino alla sezione Object Lock e scegli Modifica.
6. In Object Lock, scegli Attiva.

   Devi confermare che l'attivazione di Object Lock consentirà in modo permanente il blocco degli oggetti in questo bucket.
7. Seleziona Salvataggio delle modifiche.

Utilizzando il AWS CLI

Il comando di esempio `put-object-lock-configuration` seguente imposta un periodo di conservazione di Object Lock di 50 giorni su un bucket denominato *`example-s3-bucket1`*:

```
aws s3api put-object-lock-configuration --bucket example-s3-bucket1 --object-lock-
configuration='{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention":
 { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

Per ulteriori informazioni ed esempi, consulta put-object-lock-configuration nel Riferimento ai comandi AWS CLI .

> **Note**
>
> È possibile eseguire AWS CLI comandi dalla console utilizzando AWS CloudShell. AWS CloudShell è una shell preautenticata basata su browser che è possibile avviare direttamente da. AWS Management ConsolePer ulteriori informazioni, consulta Cos'è? CloudShell nella Guida AWS CloudShell per l'utente.

Utilizzo di REST API

Puoi utilizzare la REST API Amazon S3 per abilitare Object Lock su un bucket S3 esistente. Per ulteriori informazioni, consulta PutObjectLockConfiguration in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Utilizzo degli AWS SDK

Per esempi su come abilitare Object Lock per un bucket S3 esistente con gli AWS SDK, consulta. Utilizzo `PutObjectLockConfiguration` con un AWS SDK o una CLI

Per esempi su come ottenere la configurazione corrente di Object Lock con gli AWS SDK, consulta. Utilizzo `GetObjectLockConfiguration` con un AWS SDK o una CLI

Per uno scenario interattivo che illustra le diverse funzionalità di Object Lock utilizzando gli AWS SDK, consulta. Lavora con le funzionalità di blocco degli oggetti di Amazon S3 utilizzando un SDK AWS

Per informazioni generali sull'utilizzo di diversi AWS SDK, consulta. Sviluppo con Amazon S3 utilizzando gli SDK AWS

Impostazione o modifica di un blocco a fini legali su un oggetto S3

Puoi impostare o rimuovere un blocco legale su un oggetto S3 utilizzando la console Amazon S3 AWS CLI AWS , gli SDK o l'API REST di Amazon S3.

> **Important**
>
> - Se desideri impostare un blocco a fini legali su un oggetto, Object Lock deve già essere abilitato nel bucket dell'oggetto.
> - Quando esegui il PUT di una versione dell'oggetto che dispone di una modalità e un periodo di conservazione individuali espliciti in un bucket, le impostazioni Object Lock individuali della versione dell'oggetto hanno la precedenza su qualsiasi impostazione di conservazione delle proprietà del bucket.

Per ulteriori informazioni, consulta the section called “Blocchi a fini giudiziari”.

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket, scegli il nome del bucket contenente gli oggetti su cui desideri impostare o modificate un blocco a fini legali.

4. Nell'elenco Oggetti, seleziona l'oggetto su cui desideri impostare o modificare un blocco a fini legali.
5. Nella pagina delle proprietà dell'oggetto, individua la sezione Blocco oggetti di carattere legale e scegli Modifica.
6. Scegli Abilita per impostare un blocco a fini legali o Disabilita per rimuoverlo.
7. Seleziona Salvataggio delle modifiche.

Utilizzando il AWS CLI

L'esempio `put-object-legal-hold` seguente imposta un blocco a fini legali sull'oggetto *`my-image.fs`* nel bucket denominato *`example-s3-bucket1`*:

```
aws s3api put-object-legal-hold --bucket example-s3-bucket1 --key my-image.fs --legal-hold="Status=ON"
```

L'esempio `put-object-legal-hold` seguente rimuove un blocco a fini legali sull'oggetto *`my-image.fs`* nel bucket denominato *`example-s3-bucket1`*:

```
aws s3api put-object-legal-hold --bucket example-s3-bucket1 --key my-image.fs --legal-hold="Status=OFF"
```

Per ulteriori informazioni ed esempi, consulta put-object-legal-hold nel Riferimento ai comandi AWS CLI .

> **Note**
>
> È possibile eseguire AWS CLI comandi dalla console utilizzando AWS CloudShell. AWS CloudShell è una shell preautenticata basata su browser che è possibile avviare direttamente da. AWS Management ConsolePer ulteriori informazioni, consulta Cos'è? CloudShell nella Guida AWS CloudShell per l'utente.

Utilizzo di REST API

Puoi utilizzare la REST API per impostare o modificare un blocco a fini legali su un oggetto. Per ulteriori informazioni, consulta PutObjectLegalHold in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Utilizzo degli AWS SDK

Per esempi su come impostare un blocco legale su un oggetto con gli AWS SDK, consulta. Utilizzo `PutObjectLegalHold` con un AWS SDK o una CLI

Per alcuni esempi su come ottenere l'attuale status di conservazione legale con gli AWS SDK, consulta. Ottieni la configurazione di conservazione legale di un oggetto Amazon S3 utilizzando un SDK AWS

Per uno scenario interattivo che illustra le diverse funzionalità di Object Lock utilizzando gli AWS SDK, consulta. Lavora con le funzionalità di blocco degli oggetti di Amazon S3 utilizzando un SDK AWS

Per informazioni generali sull'utilizzo di diversi AWS SDK, consulta. Sviluppo con Amazon S3 utilizzando gli SDK AWS

Impostazione o modifica di un periodo di conservazione su un oggetto S3

Puoi impostare o modificare un periodo di conservazione su un oggetto S3 utilizzando la console Amazon S3 AWS CLI AWS , gli SDK o l'API REST di Amazon S3.

Important

- Se desideri impostare un periodo di conservazione su un oggetto, Object Lock deve già essere abilitato nel bucket dell'oggetto.
- Quando esegui il PUT di una versione dell'oggetto che dispone di una modalità e un periodo di conservazione individuali espliciti in un bucket, le impostazioni Object Lock individuali della versione dell'oggetto hanno la precedenza su qualsiasi impostazione di conservazione delle proprietà del bucket.
- L'unico modo per eliminare un oggetto in modalità di conformità prima della scadenza della data di conservazione è eliminare l'oggetto associato. Account AWS

Per ulteriori informazioni, consulta Periodi di conservazione.

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket, scegli il nome del bucket contenente l'oggetto su cui desideri impostare o modificate un periodo di conservazione.
4. Nell'elenco Oggetti, seleziona l'oggetto su cui desideri impostare o modificare un periodo di conservazione.
5. Nella pagina delle proprietà dell'oggetto, individua la sezione Conservazione del blocco oggetti e scegli Modifica.
6. In Conservazione, scegli Abilita per impostare un periodo di conservazione o Disabilita per rimuovere un periodo di conservazione.
7. Se hai scelto Abilita, in Modalità di conservazione, scegli Modalità di governance o Modalità di conformità. Per ulteriori informazioni, consulta Modalità di conservazione.
8. In Data di fine conservazione, scegli la data in cui desideri che termini il periodo di conservazione. Durante questo periodo, l'oggetto sarà protetto da WORM e non potrà essere sovrascritto o eliminato. Per ulteriori informazioni, consulta Periodi di conservazione.
9. Seleziona Save changes (Salva modifiche).

Utilizzando il AWS CLI

L'esempio `put-object-retention` seguente imposta un periodo di conservazione sull'oggetto *`my-image.fs`* nel bucket denominato *`example-s3-bucket1`* fino al 1° gennaio 2025:

```
aws s3api put-object-retention --bucket example-s3-bucket1 --key my-image.fs --
retention='{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

Per ulteriori informazioni ed esempi, consulta put-object-retention nel Riferimento ai comandi AWS CLI .

> **Note**
>
> È possibile eseguire AWS CLI comandi dalla console utilizzando AWS CloudShell. AWS CloudShell è una shell preautenticata basata su browser che è possibile avviare direttamente da. AWS Management ConsolePer ulteriori informazioni, consulta Cos'è? CloudShell nella Guida AWS CloudShell per l'utente.

Utilizzo di REST API

Puoi utilizzare la REST API per impostare un periodo di conservazione su un oggetto. Per ulteriori informazioni, consulta PutObjectRetention in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Utilizzo degli AWS SDK

Per esempi su come impostare un periodo di conservazione su un oggetto con gli AWS SDK, consulta. Utilizzo `PutObjectRetention` con un AWS SDK o una CLI

Per esempi su come ottenere il periodo di conservazione di un oggetto con gli AWS SDK, consulta. Utilizzo `GetObjectRetention` con un AWS SDK o una CLI

Per uno scenario interattivo che illustra le diverse funzionalità di Object Lock utilizzando gli AWS SDK, consulta. Lavora con le funzionalità di blocco degli oggetti di Amazon S3 utilizzando un SDK AWS

Per informazioni generali sull'utilizzo di diversi AWS SDK, consulta. Sviluppo con Amazon S3 utilizzando gli SDK AWS

Impostazione o modifica di un periodo di conservazione predefinito su un bucket S3

Puoi impostare o modificare un periodo di conservazione predefinito su un bucket S3 utilizzando la console Amazon S3 AWS CLI AWS , gli SDK o l'API REST di Amazon S3. Specifica una durata, in giorni o in anni, per stabilire quanto a lungo proteggere ogni versione di un oggetto inserita nel bucket.

Important

- Se desideri impostare un periodo di conservazione su un bucket, Object Lock deve già essere abilitato nel bucket.
- Quando esegui il PUT di una versione dell'oggetto che dispone di una modalità e un periodo di conservazione individuali espliciti in un bucket, le impostazioni Object Lock individuali della versione dell'oggetto hanno la precedenza su qualsiasi impostazione di conservazione delle proprietà del bucket.
- L'unico modo per eliminare un oggetto in modalità di conformità prima della scadenza della data di conservazione è eliminare l'oggetto associato. Account AWS

Per ulteriori informazioni, consulta Periodi di conservazione.

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket, scegli il nome del bucket su cui desideri impostare o modificate un periodo di conservazione.
4. Scegliere la scheda Properties (Proprietà).
5. In Proprietà, scorri verso il basso fino alla sezione Object Lock e scegli Modifica.
6. In Conservazione predefinita, scegli Abilita per impostare un periodo di conservazione predefinito o Disabilita per rimuovere un periodo di conservazione predefinito.
7. Se hai scelto Abilita, in Modalità di conservazione, scegli Modalità di governance o Modalità di conformità. Per ulteriori informazioni, consulta Modalità di conservazione.
8. In Periodo di conservazione predefinito, scegli il numero di giorni o anni di durata del periodo di conservazione. Gli oggetti inseriti in questo bucket verranno bloccati per questo numero di giorni o anni. Per ulteriori informazioni, consulta Periodi di conservazione.
9. Seleziona Save changes (Salva modifiche).

Utilizzando il AWS CLI

Il comando di esempio `put-object-lock-configuration` seguente imposta un periodo di conservazione di Object Lock di 50 giorni su un bucket denominato *`example-s3-bucket1`* utilizzando la modalità di conformità:

```
aws s3api put-object-lock-configuration --bucket example-s3-bucket1 --object-lock-
configuration='{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention":
 { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

L'esempio `put-object-lock-configuration` seguente rimuove la configurazione di conservazione predefinita su un bucket:

```
aws s3api put-object-lock-configuration --bucket example-s3-bucket1 --object-lock-
configuration='{ "ObjectLockEnabled": "Enabled"}'
```

Per ulteriori informazioni ed esempi, consulta put-object-lock-configuration nel Riferimento ai comandi AWS CLI .

> **Note**
>
> È possibile eseguire AWS CLI comandi dalla console utilizzando AWS CloudShell. AWS CloudShell è una shell preautenticata basata su browser che è possibile avviare direttamente da. AWS Management ConsolePer ulteriori informazioni, consulta Cos'è? CloudShell nella Guida AWS CloudShell per l'utente.

Utilizzo di REST API

Puoi utilizzare l'API REST per impostare un periodo di conservazione predefinito su un bucket S3 esistente. Per ulteriori informazioni, consulta PutObjectLockConfiguration in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Utilizzo degli SDK AWS

Per esempi su come impostare un periodo di conservazione predefinito su un bucket S3 esistente con gli AWS SDK, consulta. Utilizzo `PutObjectLockConfiguration` con un AWS SDK o una CLI

Per uno scenario interattivo che illustra le diverse funzionalità di Object Lock utilizzando gli SDK, consulta AWS . Lavora con le funzionalità di blocco degli oggetti di Amazon S3 utilizzando un SDK AWS

Per informazioni generali sull'utilizzo di diversi AWS SDK, consulta. Sviluppo con Amazon S3 utilizzando gli SDK AWS

Utilizzo delle classi di storage di Amazon S3

A ogni oggetto di Amazon S3 è associata una classe di storage. Ad esempio, se si elencano tutti gli oggetti in un bucket S3, la console mostra la classe di storage di tutti gli oggetti nell'elenco. In Amazon S3 è disponibile una gamma di classi di storage per gli oggetti che vengono archiviati dall'utente. Puoi scegliere una classe di storage a seconda dello scenario del caso d'uso e dei requisiti relativi all'accesso e alle prestazioni. Tutte queste classi di storage offrono un livello elevato di durabilità.

Nelle sezioni seguenti vengono fornite informazioni dettagliate sulle varie classi di storage e su come impostare la classe di storage più adatta ai tuoi oggetti.

Argomenti

Classi di storage per oggetti a cui si accede di frequente

Per i casi d'uso sensibili alle prestazioni (quelli che richiedono un tempo di accesso in millisecondi) e per i dati a cui si accede di frequente, Amazon S3 fornisce le seguenti classi di storage:

- S3 Standard: classe di archiviazione predefinita. Se al momento del caricamento di un oggetto non specifichi una classe di storage, Amazon S3 assegna la classe di storage S3 Standard.
- S3 Express One Zone: Amazon S3 Express One Zone è una classe di archiviazione Amazon S3 a zona singola ad alte prestazioni, progettata appositamente per fornire un accesso ai dati coerente di pochi millisecondi per applicazioni sensibili alla latenza. S3 Express One Zone è la classe di cloud object storage con la latenza più bassa disponibile oggi, con una velocità di accesso ai dati fino a 10 volte più veloce e con costi di richiesta inferiori del 50% rispetto a S3 Standard. Con S3 Express One Zone, i dati vengono archiviati in modo ridondante su più dispositivi all'interno di una singola zona di disponibilità. Per ulteriori informazioni, consulta Che cos'è S3 Express One Zone?.
- Reduced Redundancy: la classe di archiviazione Reduced Redundancy Storage (RRS) è concepita per dati riproducibili non critici che possono essere archiviati con una ridondanza inferiore rispetto alla classe di archiviazione S3 Standard.

> **Important**
>
> È consigliabile non utilizzare questa classe di archiviazione. La classe di archiviazione S3 Standard è più conveniente in termini di costi.

Riguardo la durabilità, gli oggetti RRS hanno una perdita di oggetti annua media stimata dello 0,01%. Se perdi un oggetto RRS, Amazon S3 restituisce un errore 405 per le richieste eseguite a tale oggetto.

Classe di storage per ottimizzare automaticamente i dati con modelli di accesso variabili o sconosciuti

S3 Intelligent-Tiering (Piano intelligente S3) è una classe di archiviazione di Amazon S3 progettata per ottimizzare i costi di archiviazione spostando automaticamente i dati sul livello di accesso più conveniente, senza impatto sulle prestazioni o sul sovraccarico operativo. S3 Intelligent-Tiering (Piano intelligente S3) è l'unica classe di archiviazione cloud in grado di offrire risparmi automatici sui costi spostando i dati a livello granulare degli oggetti tra i livelli di accesso quando i modelli di accesso cambiano. S3 Intelligent-Tiering (Piano intelligente S3) è la classe di archiviazione ideale per chi vuole ottimizzare i costi di archiviazione per i dati con modelli di accesso sconosciuti o variabili. Non sono previste spese di recupero per S3 Intelligent-Tiering.

Per una tariffa mensile ridotta di monitoraggio degli oggetti e di automazione, S3 Intelligent-Tiering (Piano intelligente S3) monitora i modelli di accesso e sposta automaticamente gli oggetti a cui non è stato eseguito l'accesso a livelli più convenienti in termini di costi. S3 Intelligent-Tiering offre risparmi automatici sui costi di archiviazione in tre livelli di accesso a bassa latenza ed elevata velocità di trasmissione effettiva. Per i dati a cui è possibile accedere in modo asincrono, puoi scegliere di attivare le funzionalità di archiviazione automatica all'interno della classe di archiviazione S3 Intelligent-Tiering. S3 Intelligent-Tiering è progettato per una disponibilità del 99,9% e una durata del 99,999999999%.

La classe di archiviazione S3 Intelligent-Tiering (Piano intelligente S3) archivia automaticamente gli oggetti in tre livelli di accesso.

- Frequent Access (Accesso frequente): gli oggetti caricati o trasferiti nella classe S3 Intelligent-Tiering (Piano intelligente S3) vengono archiviati automaticamente nel livello Frequent Access (Accesso frequente).
- Infrequent Access (Accesso infrequente): S3 Intelligent-Tiering (Piano intelligente S3) sposta gli oggetti a cui non è stato eseguito l'accesso per 30 giorni consecutivi al livello Infrequent Access (Accesso infrequente).

- Archive Instant Access (Archiviazione con accesso istantaneo): con S3 Intelligent-Tiering (Piano intelligente S3), tutti gli oggetti esistenti a cui non è stato eseguito l'accesso per 90 giorni consecutivi si sposteranno automaticamente al livello Archive Instant Access (Archiviazione con accesso istantaneo).

Oltre a questi tre livelli, S3 Intelligent-Tiering (Piano intelligente S3) offre due livelli facoltativi di accesso all'archiviazione:

- Archive Access (Accesso archiviazione): S3 Intelligent-Tiering (Piano intelligente S3) offre la possibilità di attivare il livello Archive Access (Accesso archiviazione) per i dati a cui è possibile accedere in modo asincrono. Dopo l'attivazione, il livello Archive Access archivia automaticamente gli oggetti a cui non è stato eseguito l'accesso per un minimo di 90 giorni consecutivi.
- Deep Archive Access (Accesso archiviazione profonda): S3 Intelligent-Tiering (Piano intelligente S3) offre la possibilità di attivare il livello Deep Archive Access (Accesso archiviazione profonda) per i dati a cui è possibile accedere in modo asincrono. Dopo l'attivazione, il livello Deep Archive Access archivia automaticamente gli oggetti a cui non è stato eseguito l'accesso per un minimo di 180 giorni consecutivi.

> **Note**
>
> - Attivare il livello Archive Access per 90 giorni solo se si desidera ignorare il livello Archive Instant Access. Il livello Archive Access offre uno storage a costi leggermente inferiori con tempi di recupero. minute-to-hour Il livello Archive Instant Access (Archiviazione con accesso istantaneo) offre un accesso in millisecondi e prestazioni a elevata velocità di trasmissione effettiva.
> - Attiva i livelli Accesso di archiviazione e Accesso di archiviazione profonda solo se l'applicazione può accedere agli oggetti in modo asincrono. Se l'oggetto recuperato è archiviato nei livelli Archive Access (Accesso archiviazione) o Deep Archive Access (Accesso archiviazione profondo), prima devi ripristinarlo utilizzando `RestoreObject`.

Puoi spostare i nuovi dati creati alla classe di archiviazione Piano intelligente S3, impostandola come classe di archiviazione predefinita. Puoi anche scegliere di attivare uno o entrambi i livelli di accesso all'archivio utilizzando il funzionamento dell'`PutBucketIntelligentTieringConfiguration`API AWS CLI, la o la console Amazon S3. Per ulteriori informazioni sull'utilizzo di S3 Intelligent-Tiering

(Piano intelligente S3) e sull'attivazione dei livelli di accesso all'archiviazione, consulta Utilizzare S3 Intelligent-Tiering.

Per accedere agli oggetti nei livelli Accesso archivio o Accesso archivio approfondito, devi prima ripristinarli. Per ulteriori informazioni, consulta Ripristino degli oggetti dai livelli Archive Access e Deep Archive Access di S3 Intelligent-Tiering.

> **Note**
>
> Se le dimensioni di un oggetto sono inferiori a 128 KB, questo non è monitorato e il tiering automatico non è consentito. Gli oggetti più piccoli vengono sempre archiviati nel livello Accesso frequente. Per ulteriori informazioni su S3 Intelligent-Tiering (Piano intelligente S3), consulta Livelli di accesso S3 Intelligent-Tiering.

Classi di storage per oggetti a cui si accede raramente

Le classi di archiviazione S3 Standard-IA e S3 One Zone-IA sono concepite per dati di lunga durata e ai quali si accede raramente. IA è l'acronimo di Infrequent Access (accesso non frequente). Gli oggetti S3 Standard-IA e S3 One Zone-IA sono disponibili per l'accesso in millisecondi (simile alla classe di archiviazione S3 Standard). Amazon S3 addebita un costo per il recupero di questi oggetti, di conseguenza sono più appropriati per i dati a cui si accede raramente. Per informazioni sui prezzi, consulta Prezzi di Amazon S3.

Ad esempio, potresti scegliere le classi di storage S3 Standard-IA e S3 One Zone-IA:

- Per lo storage di backup.
- Per i dati più vecchi a cui si accede raramente ma che richiedono l'accesso in millisecondi. Ad esempio, quando carichi i dati, potresti scegliere la classe di archiviazione S3 Standard e utilizzare la configurazione del ciclo di vita per indicare ad Amazon S3 di eseguire la transizione degli oggetti alla classe S3 Standard-IA o S3 One Zone-IA.

  Per ulteriori informazioni sulla gestione del ciclo di vita, consulta Gestione del ciclo di vita dello storage.

> **Note**
>
> Le classi di storage S3 Standard-IA e S3 One Zone-IA sono ideali per gli oggetti di dimensioni superiori a 128 KB che desideri conservare per almeno 30 giorni. Se un oggetto è inferiore a 128 KB, Amazon S3 addebita il costo relativo a 128 KB. Se elimini un oggetto prima della fine del periodo minimo di storage di 30 giorni, viene addebitato un costo corrispondente a 30 giorni. Gli oggetti eliminati, sovrascritti o trasferiti a una classe di archiviazione diversa prima di 30 giorni sono soggetti al normale costo di utilizzo dell'archiviazione e all'addebito ripartito proporzionalmente per il resto del periodo minimo di 30 giorni. Per informazioni sui prezzi, consulta Prezzi di Amazon S3.

Di seguito sono riportate le differenze tra queste classi di storage:

- S3 Standard-IA: Amazon S3 archivia i dati dell'oggetto in modo ridondante su più zone di disponibilità geograficamente separate (simile alla classe di archiviazione S3 Standard). Gli oggetti S3 Standard-IA sono resilienti alla perdita di una zona di disponibilità. Questa classe di archiviazione offre una maggiore disponibilità e resilienza rispetto alla classe S3 One Zone-IA.
- S3 One Zone-IA: Amazon S3 archivia i dati dell'oggetto in una sola zona di disponibilità e il costo è quindi inferiore rispetto alla classe S3 Standard-IA. Tuttavia, i dati non sono resilienti alla perdita fisica della zona di disponibilità dovuta a disastri naturali, come terremoti e alluvioni. La classe di archiviazione S3 One Zone-IA è durevole quanto la classe S3 Standard-IA, ma è meno disponibile e meno resiliente. Per un confronto della durabilità e della disponibilità delle classi di storage, consulta Confronto delle classi di storage di Amazon S3 alla fine della sezione. Per informazioni sui prezzi, consulta Prezzi di Amazon S3.

Consigliamo quanto segue:

- S3 Standard-IA: da utilizzare per la copia principale o l'unica copia dei dati che non può essere ricreata.
- S3 One Zone-IA: da utilizzare quando è possibile ricreare i dati in caso di problemi con la zona di disponibilità e per le repliche di oggetti quando si imposta la replica tra regioni (CRR) S3.

Classi di archiviazione per oggetti a cui si accede raramente

Le classi di storage S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive sono progettate per lo storage e l'archiviazione dei dati a basso costo e a lungo termine. Queste classi di archiviazione offrono la stessa durabilità e resilienza della classe di archiviazione S3 Standard e S3 Standard-IA. Per ulteriori informazioni sulle classi di storage S3 Glacier, consulta. Archiviazione dei dati a lungo termine utilizzando le classi di storage S3 Glacier

Amazon S3 fornisce le seguenti classi di storage S3 Glacier:

- S3 Glacier Instant Retrieval: da utilizzare per dati a lungo termine a cui si accede raramente e che richiedono un recupero di millisecondi. I dati di questa classe di archiviazione sono disponibili per l'accesso in tempo reale.
- S3 Glacier Flexible Retrieval (Recupero flessibile S3 Glacier): da utilizzare per le archiviazioni con porzioni di dati da recuperare in pochi minuti. I dati di questa classe di archiviazione sono archiviati e non sono disponibili per l'accesso in tempo reale.
- S3 Glacier Deep Archive (Archiviazione profonda S3 Glacier): utilizzata per l'archiviazione di dati a cui è necessario accedere raramente. I dati di questa classe di archiviazione sono archiviati e non sono disponibili per l'accesso in tempo reale.

Recupero di oggetti archiviati

Puoi impostare la classe di archiviazione di un oggetto su S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive come per qualsiasi altra classe di archiviazione, come descritto nella sezione Impostazione della classe di storage di un oggetto. Tuttavia, gli oggetti S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive sono archiviati e non sono disponibili per l'accesso in tempo reale. Per ulteriori informazioni, consulta Archiviazione.

> **Note**
>
> Quando usi le classi di storage S3 Glacier, i tuoi oggetti rimangono in Amazon S3. Non puoi accedervi direttamente tramite il servizio Amazon S3 Glacier separato. Per informazioni sul servizio Amazon S3 Glacier, consulta la Amazon S3 Glacier Developer Guide.

Classe di storage per Amazon S3 su Outposts

Con Amazon S3 on Outposts, puoi creare bucket S3 sulle tue AWS Outposts risorse e archiviare e recuperare oggetti in locale per applicazioni che richiedono l'accesso locale ai dati, l'elaborazione locale dei dati e la residenza dei dati. Puoi utilizzare le stesse operazioni e funzionalità API di Amazon S3, tra cui policy di accesso, crittografia e tagging. AWS Outposts Puoi usare S3 su Outposts tramite AWS CLI, AWS , SDK o AWS Management Console l'API REST.

S3 su Outposts offre una nuova classe di storage, S3 Outposts (`OUTPOSTS`). La classe di archiviazione S3 Outposts è disponibile solo per gli oggetti archiviati in bucket su Outposts. Se tenti di utilizzare questa classe di archiviazione con un bucket S3 in un, si verifica un Regione AWS errore. `InvalidStorageClass` Inoltre, se provi a utilizzare altre classi di storage S3 con oggetti archiviati in bucket S3 su Outposts, si avrà la stessa risposta di errore.

Gli oggetti archiviati nella classe di storage S3 Outposts (`OUTPOSTS`) vengono crittografati sempre utilizzando la crittografia lato server con chiavi di crittografia gestite di Amazon S3 (SSE-S3). Per ulteriori informazioni, consulta Uso della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).

Puoi inoltre scegliere di crittografare esplicitamente gli oggetti archiviati nella classe di storage S3 Outposts utilizzando la crittografia lato server con chiavi di crittografia fornite dal cliente (SSE-C). Per ulteriori informazioni, consulta Utilizzo della crittografia lato server con chiavi fornite dal cliente (SSE-C).

> **Note**
>
> S3 on Outposts non supporta la crittografia lato server AWS Key Management Service con chiavi AWS KMS() (SSE-KMS).

Per ulteriori informazioni su S3 su Outposts, consulta Che cos'è Amazon S3 su Outposts?.

Confronto delle classi di storage di Amazon S3

Nella tabella seguente vengono confrontate le classi di storage con disponibilità, durata, durata minima di storage e altre considerazioni.

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

* Recupero flessibile S3 Glacier richiede 40 KB di metadati aggiuntivi per ogni oggetto archiviato. Ciò include 32 KB di metadati addebitati alla tariffa Recupero flessibile S3 Glacier (richiesta per identificare e recuperare i dati) e altri 8 KB di dati addebitati alla tariffa S3 Standard. La tariffa S3 Standard è necessaria per mantenere il nome e i metadati definiti dall'utente per gli oggetti archiviati in Recupero flessibile S3 Glacier. Per ulteriori informazioni sulle classi di storage, consultare Classi di storage di Amazon S3.

** S3 Glacier Deep Archive richiede 40 KB di metadati aggiuntivi per ogni oggetto archiviato. Ciò include 32 KB di metadati addebitati alla tariffa Deep Archive Amazon S3 Glacier (richiesta per identificare e recuperare i dati) e altri 8 KB di dati addebitati alla tariffa S3 Standard. La tariffa S3 Standard è necessaria per mantenere il nome e i metadati definiti dall'utente per gli oggetti archiviati in Deep Archive Amazon S3 Glacier. Per ulteriori informazioni sulle classi di storage, consultare Classi di storage di Amazon S3.

Tieni presente che tutte le classi di archiviazione, fatta eccezione per S3 One Zone-IA e S3 Express One Zone, sono progettate per essere resilienti a perdite fisiche di una zona di disponibilità causate da calamità. Oltre ai requisiti relativi alle prestazioni, devi considerare anche i costi. Per il prezzo delle classi di storage, consulta Prezzi di Amazon S3.

Impostazione della classe di storage di un oggetto

Per impostare e aggiornare le classi di storage degli oggetti, puoi utilizzare la console Amazon S3, AWS gli SDK o (). AWS Command Line Interface AWS CLI Tutti questi approcci utilizzano le operazioni API di Amazon S3 per inviare richieste ad Amazon S3.

Le operazioni API di Amazon S3 supportano l'impostazione (o l'aggiornamento) della classe di archiviazione degli oggetti come segue:

- Alla creazione di un nuovo oggetto, è possibile specificarne la relativa classe di storage. Ad esempio, quando si creano oggetti tramite le operazioni API PUT Object, POST Object e Initiate Multipart Upload, aggiungi la richiesta `x-amz-storage-class` per specificare la classe di archiviazione. Se non aggiungi questa intestazione, Amazon S3 utilizza la classe di archiviazione predefinita S3 Standard.
- È anche possibile modificare la classe di archiviazione di un oggetto già archiviato in Amazon S3 in un'altra classe di archiviazione creando una copia dell'oggetto tramite l'operazione API PUT Object - Copy. Tuttavia, non è possibile utilizzare PUT Object - Copy per copiare oggetti archiviati nelle classi di archiviazione S3 Glacier Flexible Retrieval (Recupero flessibile S3 Glacier) o S3 Glacier Deep Archive (Archiviazione profonda S3 Glacier). Non è inoltre possibile passare da S3 One Zone-IA a S3 Glacier Instant Retrieval.

  Puoi copiare l'oggetto nello stesso bucket utilizzando lo stesso nome di chiave e specificando le intestazioni delle richieste come segue:

  - Imposta l'intestazione `x-amz-metadata-directive` su `COPY`.
  - Imposta l'intestazione `x-amz-storage-class` sulla classe di archiviazione che desideri usare.

  In un bucket abilitato per il controllo delle versioni, non puoi modificare la classe di storage di una versione specifica di un oggetto. Al momento della copia, Amazon S3 assegna all'oggetto un nuovo ID versione.
- Puoi modificare la classe di storage di un oggetto utilizzando la console Amazon S3 se la dimensione dell'oggetto è inferiore a 160 GB. Se è più grande, si consiglia di aggiungere la configurazione del ciclo di vita di S3 per modificare la classe di storage dell'oggetto.
- Se utilizzi la console Amazon S3 per modificare la classe di storage per un oggetto con tag definiti dall'utente, devi disporre dell'autorizzazione. `s3:GetObjectTagging` Se stai modificando la classe di archiviazione per un oggetto che non ha tag definiti dall'utente ma ha una dimensione superiore a 16 MB, devi disporre anche dell'autorizzazione. `s3:GetObjectTagging` Se la policy del bucket di destinazione nega l'`s3:GetObjectTagging`azione, la classe di archiviazione per l'oggetto verrà aggiornata, ma i tag definiti dall'utente verranno rimossi dall'oggetto e verrà visualizzato un errore.
- È possibile indicare a Amazon S3 di modificare la classe di storage degli oggetti aggiungendo la configurazione del ciclo di vita di S3 a un bucket. Per ulteriori informazioni, consulta Gestione del ciclo di vita dello storage.

- Quando imposti la configurazione della replica, puoi impostare la classe di storage per gli oggetti replicati su qualsiasi altra classe di storage. Tuttavia, non è possibile copiare oggetti archiviati nelle classi di archiviazione S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive. Per ulteriori informazioni, consulta Configurazione di replica.

Limitazione delle autorizzazioni delle policy di accesso a una classe di storage specifica

Quando concedi le autorizzazioni alle policy di accesso per le operazioni Amazon S3, è possibile utilizzare la chiave di condizione `s3:x-amz-storage-class` per limitare la classe di storage da utilizzare durante l'archiviazione degli oggetti caricati. Ad esempio, quando concedi l'autorizzazione `s3:PutObject`, puoi limitare il caricamento di oggetti a una classe di archiviazione specifica. Per un esempio di policy, consulta Esempio: limitazione dei caricamenti di oggetti a oggetti con una classe di archiviazione specifica.

Per ulteriori informazioni sull'utilizzo delle condizioni nelle policy e per l'elenco completo delle chiavi di condizione Amazon S3, consulta i seguenti argomenti:

- Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference
- Esempi di policy Bucket che utilizzano chiavi condizionali

Archiviazione dei dati a lungo termine utilizzando le classi di storage S3 Glacier

Amazon S3 offre diverse classi di storage S3 Glacier progettate per fornire soluzioni convenienti per l'archiviazione di dati a lungo termine a cui non si accede spesso. Le classi di storage S3 Glacier sono:

- S3 Glacier Instant Retrieval
- S3 Glacier Flexible Retrieval
- S3 Glacier Deep Archive

Scegli una di queste classi di storage in base alla frequenza con cui accedi ai dati e alla velocità con cui devi recuperarli. Ognuna di queste classi di storage offre la stessa durata e resilienza della classe di storage S3 Standard, ma a costi di storage inferiori. Per ulteriori informazioni sulle classi di storage S3 Glacier, consulta https://aws.amazon.com/s3/storage-classes/glacier/.

Argomenti

Confronto delle classi di storage S3 Glacier

Ogni classe di storage S3 Glacier ha una durata di archiviazione minima per tutti gli oggetti. Se elimini, sovrascrivi o trasferisci l'oggetto a una classe di archiviazione diversa prima del minimo, ti viene addebitata l'intera durata minima di archiviazione.

Alcune classi di storage S3 Glacier sono archiviate, il che significa che gli oggetti archiviati in tali classi sono archiviati e non sono disponibili per l'accesso in tempo reale. Per ulteriori informazioni, consulta Archiviazione.

Le classi di storage progettate per modelli di accesso meno frequenti con tempi di recupero più lunghi offrono costi di storage inferiori. Per informazioni sui prezzi, consulta. https://aws.amazon.com/s3/pricing/

La tabella seguente riassume i punti chiave da considerare nella scelta di una classe di storage S3 Glacier:

S3 Glacier Instant Retrieval

Consigliamo di utilizzare S3 Glacier Instant Retrieval per dati a lungo termine a cui si accede una volta al trimestre e che richiedono tempi di recupero di millisecondi. Questa classe di archiviazione è ideale per casi d'uso sensibili alle prestazioni come l'hosting di immagini, le applicazioni di condivisione di file e l'archiviazione di cartelle cliniche per l'accesso durante gli appuntamenti.

La classe di storage S3 Glacier Instant Retrieval offre accesso in tempo reale agli oggetti con le stesse prestazioni di latenza e throughput della classe di storage S3 Standard-IA. Rispetto a S3 Standard-IA, S3 Glacier Instant Retrieval ha costi di storage inferiori ma costi di accesso ai dati più elevati.

Esiste una dimensione minima dell'oggetto di 128 KB per i dati archiviati nella classe di storage S3 Glacier Instant Retrieval. Questa classe di archiviazione ha anche una durata minima di archiviazione di 90 giorni.

S3 Glacier Flexible Retrieval

Consigliamo di utilizzare S3 Glacier Flexible Retrieval per archiviare i dati a cui si accede una o due volte all'anno e che non richiedono un accesso immediato. S3 Glacier Flexible Retrieval offre tempi di recupero flessibili per aiutarti a bilanciare i costi, con tempi di accesso che vanno da pochi minuti a ore e recuperi di massa gratuiti. Questa classe di storage è ideale per il backup e il disaster recovery.

Gli oggetti archiviati in S3 Glacier Flexible Retrieval sono archiviati e non sono disponibili per l'accesso in tempo reale. Per ulteriori informazioni, consulta Archiviazione. Per accedere a questi oggetti, devi prima avviare una richiesta di ripristino che crea una copia temporanea dell'oggetto a cui puoi accedere una volta completata la richiesta. Per informazioni, consulta Utilizzo di oggetti archiviati. Quando ripristini un oggetto, puoi scegliere un livello di recupero adatto al tuo caso d'uso, con costi inferiori per tempi di ripristino più lunghi.

I seguenti livelli di recupero sono disponibili per S3 Glacier Flexible Retrieval:

- Recupero rapido: in genere ripristina l'oggetto in 1-5 minuti. I recuperi rapidi sono soggetti alla richiesta, quindi per garantire tempi di ripristino affidabili e prevedibili, si consiglia di acquistare una capacità di recupero predisposta. Per ulteriori informazioni, consulta Capacità con provisioning.
- Recupero standard: in genere ripristina l'oggetto in 3-5 ore o entro 1-5 ore se si utilizza S3 Batch Operations. Per ulteriori informazioni, consulta Ripristino di oggetti con operazioni in batch.
- Recupero in blocco: in genere ripristina l'oggetto entro 5-12 ore. I recuperi in blocco sono gratuiti.

La durata minima di archiviazione per gli oggetti nella classe di storage S3 Glacier Flexible Retrieval è di 90 giorni.

S3 Glacier Flexible Retrieval richiede 40 KB di metadati aggiuntivi per ogni oggetto. Ciò include 32 KB di metadati necessari per identificare e recuperare i dati, che vengono addebitati alla tariffa predefinita per S3 Glacier Flexible Retrieval. Sono necessari altri 8 KB di dati per mantenere il nome e i metadati definiti dall'utente per gli oggetti archiviati e vengono addebitati alla tariffa S3 Standard.

S3 Glacier Deep Archive

Ti consigliamo di utilizzare S3 Glacier Deep Archive per archiviare i dati a cui si accede meno di una volta all'anno. Questa classe di storage è progettata per conservare i set di dati per più anni per

soddisfare i requisiti di conformità e può essere utilizzata anche per il backup o il disaster recovery o per qualsiasi dato a cui si accede raramente e il cui recupero può richiedere fino a 72 ore. Deep Archive Amazon S3 Glacier è l'opzione di archiviazione più conveniente di AWS.

Gli oggetti archiviati in S3 Glacier Deep Archive sono archiviati e non sono disponibili per l'accesso in tempo reale. Per ulteriori informazioni, consulta Archiviazione. Per accedere a questi oggetti, devi prima avviare una richiesta di ripristino che crea una copia temporanea dell'oggetto a cui puoi accedere una volta completata la richiesta. Per informazioni, consulta Utilizzo di oggetti archiviati. Quando ripristini un oggetto, puoi scegliere un livello di recupero adatto al tuo caso d'uso, con costi inferiori per tempi di ripristino più lunghi.

I seguenti livelli di recupero sono disponibili per S3 Glacier Deep Archive:

- Recupero standard: in genere ripristina l'oggetto entro 12 ore o entro 9-12 ore quando si utilizza S3 Batch Operations. Per ulteriori informazioni, consulta Ripristino di oggetti con operazioni in batch.
- Recupero in blocco: in genere ripristina l'oggetto entro 48 ore a una frazione del costo del livello di recupero Standard.

La durata minima di archiviazione per gli oggetti nella classe di storage S3 Glacier Deep Archive è di 180 giorni.

S3 Glacier Deep Archive richiede 40 KB di metadati aggiuntivi per ogni oggetto. Ciò include 32 KB di metadati necessari per identificare e recuperare i dati, addebitati alla tariffa predefinita per S3 Glacier Deep Archive. Sono necessari altri 8 KB di dati per mantenere il nome e i metadati definiti dall'utente per gli oggetti archiviati e vengono addebitati alla tariffa S3 Standard.

Archiviazione

S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive sono classi di archiviazione. Ciò significa che quando si archivia un oggetto in queste classi di archiviazione, tale oggetto viene archiviato e non è possibile accedervi direttamente. Per accedere a un oggetto archiviato, si invia una richiesta di ripristino dell'oggetto e si attende che il servizio ripristini l'oggetto. La richiesta di ripristino ripristina una copia temporanea dell'oggetto e tale copia viene eliminata allo scadere della durata specificata nella richiesta. Per ulteriori informazioni, consulta Utilizzo di oggetti archiviati.

Queste classi di archiviazione richiedono 40 KB di metadati aggiuntivi per ogni oggetto archiviato. Ciò include 32 KB di metadati necessari per identificare e recuperare i dati, che vengono addebitati alla tariffa predefinita per quella classe di storage. Sono necessari altri 8 KB di dati per mantenere il nome e i metadati definiti dall'utente per gli oggetti archiviati e vengono addebitati alla tariffa S3 Standard.

Gli oggetti di queste classi di archiviazione vengono fatturati alle tariffe delle classi di storage S3 Standard quando vengono caricati utilizzando caricamenti multiparte. Per ulteriori informazioni, consulta Caricamento in più parti e prezzi.

È possibile ripristinare gli oggetti archiviati in queste classi di archiviazione con un massimo di 1.000 transazioni al secondo (TPS) di richieste di ripristino degli oggetti per account. Regione AWS

In che modo queste classi di storage differiscono dal servizio S3 Glacier

Le classi di storage S3 Glacier fanno parte del servizio Amazon S3 e archiviano i dati come oggetti nei bucket S3. Puoi gestire gli oggetti in queste classi di storage utilizzando la console S3 o a livello di codice utilizzando le API o gli SDK S3. Quando archivi oggetti nelle classi di storage S3 Glacier, puoi utilizzare le funzionalità di S3 come la crittografia avanzata, il tagging degli oggetti e le configurazioni del ciclo di vita di S3 per aiutare a gestire l'accessibilità e i costi dei dati.

> **Important**
>
> Ti consigliamo di utilizzare le classi di storage S3 Glacier all'interno del servizio Amazon S3 per tutti i tuoi dati a lungo termine.

Il servizio Amazon S3 Glacier (S3 Glacier) è un servizio separato che archivia i dati come archivi all'interno di casseforti. Questo servizio non supporta le funzionalità di Amazon S3 e non fornisce supporto da console per le operazioni di caricamento e download dei dati. Non è consigliabile utilizzare il servizio S3 Glacier per i dati a lungo termine. I dati archiviati in questo servizio non sono accessibili dal servizio Amazon S3. Se stai cercando informazioni sul servizio S3 Glacier, consulta la Amazon S3 Glacier Developer Guide. Per trasferire dati dal servizio Amazon S3 Glacier a una classe di storage in Amazon S3, consulta Data Transfer from Amazon S3 Glacier Vaults ad Amazon S3 nella libreria delle soluzioni. AWS

Amazon S3 Intelligent-Tiering

La classe di archiviazione S3 Intelligent-Tiering è progettata per ottimizzare i costi di archiviazione spostando automaticamente i dati al livello di accesso più conveniente quando i modelli di accesso subiscono cambiamenti, senza impatto sulle prestazioni o sovraccarico operativo. Per un monitoraggio degli oggetti mensile e una tariffa di automazione bassi, S3 Intelligent-Tiering monitora i modelli di accesso e sposta automaticamente gli oggetti ai quali non è stato eseguito l'accesso a livelli di accesso a costo più basso.

S3 Intelligent-Tiering offre risparmi automatici sui costi di archiviazione in tre livelli di accesso a bassa latenza ed elevata velocità effettiva. Per i dati a cui è possibile accedere in modo asincrono, puoi scegliere di attivare le funzionalità di archiviazione automatica all'interno della classe di archiviazione S3 Intelligent-Tiering. Non sono previste spese di recupero in S3 Intelligent-Tiering. Se si accede in seguito a un oggetto nel livello Accesso Infrequente o Accesso Istantaneo all'Archivio, l'oggetto verrà automaticamente spostato nel livello Accesso Frequente. Lo spostamento di oggetti tra i livelli di accesso all'interno della classe di archiviazione S3 Intelligent-Tiering non comporta l'applicazione di costi di livello.

S3 Intelligent-Tiering è la classe di archiviazione consigliata per i dati con modelli di accesso sconosciuti, in mutamento o imprevedibili, indipendentemente dalla dimensione dell'oggetto o dal periodo di conservazione, come data lake, analisi dei dati e nuove applicazioni.

Per informazioni sull'utilizzo di S3 Intelligent-Tiering, consulta le sezioni seguenti:

Argomenti

Come funziona S3 Intelligent-Tiering

La classe di archiviazione Amzon S3 Intelligent-Tiering memorizza automaticamente gli oggetti in tre livelli di accesso. Un livello è ottimizzato per l'accesso frequente, un livello a basso costo è ottimizzato per l'accesso infrequente e un altro livello a costi minimi è ottimizzato per i dati a cui si accede raramente. Per una ridotta tariffa mensile per l'automazione e il monitoraggio degli oggetti, S3 Intelligent-Tiering monitora i modelli di accesso e sposta automaticamente gli oggetti a cui non è stato eseguito l'accesso per 30 giorni consecutivi al livello Accesso infrequente. Dopo 90 giorni senza che sia stato eseguito l'accesso, gli oggetti vengono spostati nel livello Archive Instant Access senza impatto sulle prestazioni o sovraccarico operativo.

Per ottenere il minor costo di archiviazione sui dati a cui è possibile accedere in pochi minuti o ore, attiva funzionalità di archiviazione per avere due livelli aggiuntivi. È possibile spostare gli oggetti al livello Archive Access (Accesso archiviazione), al livello Deep Archive Access (Accesso archiviazione profonda) o a entrambi. Con il livello Archive Access (Accesso archiviazione), la classe di archiviazione S3 Intelligent-Tiering (Piano intelligente S3) sposta gli oggetti a cui non è stato eseguito l'accesso per un minimo di 90 giorni consecutivi al livello Archive Access (Accesso

archiviazione). Con il livello Deep Archive Access (Accesso archiviazione profonda), la classe di archiviazione S3 Intelligent-Tiering (Piano intelligente S3) sposta gli oggetti al livello Deep Archive Access (Accesso archiviazione profonda) dopo un minimo di 180 giorni consecutivi senza accesso. Per entrambi i livelli, puoi configurare il numero di giorni senza accesso in base alle tue esigenze.

Le seguenti azioni costituiscono un accesso che impedisce la suddivisione degli oggetti al livello Archive Access o al livello Deep Archive Access:

- Scarica o copia un oggetto archiviato tramite la console di Amazon S3.
- Richiamare CopyObject, UploadPartCopy o replicare oggetti con S3 Batch Replication. In questi casi, gli oggetti di origine delle operazioni di copia o replica sono suddivisi su più livelli.
- Invocare GetObject, PutObject, RestoreObject, CompleteMultipartUpload, ListParts o SelectObjectContent.

Ad esempio, se si accede agli oggetti tramite `SelectObjectContent` prima del numero di giorni di inattività specificato (ad esempio 180 giorni), tale azione ripristina il timer. I tuoi oggetti non passeranno al livello Archive Access o al livello Deep Archive Access fino al raggiungimento del numero di giorni specificato dopo l'ultima richiesta `SelectObjectContent`.

Se si accede in seguito a un oggetto nel livello Accesso Infrequente o Archive Instant Access, l'oggetto verrà automaticamente spostato nel livello Accesso Frequente.

Le seguenti azioni costituiscono l'accesso che automaticamente sposta gli oggetti dal livello Infrequent Access al livello Archive Instant Access e poi di nuovo al livello Frequent Access:

- Scarica o copia un oggetto tramite la console di Amazon S3.
- Invocare CopyObject, UploadPartCopy o replicare oggetti con Batch Replication. In questi casi, gli oggetti di origine delle operazioni di copia o replica sono suddivisi su più livelli.
- Invocare GetObject, PutObject, RestoreObject, CompleteMultipartUpload o ListParts.

Le altre azioni non costituiscono un accesso che automaticamente muove gli oggetti dal livello Infrequent Access o Archive Instant Access di nuovo al livello Frequent Access. Di seguito è riportato un esempio, non un elenco definitivo, di tali azioni:

- Invocare HeadObject, GetObjectTagging, PutObjectTagging, ListObjects, ListObjectsV2 o ListObjectVersions.

- L'invocazione di SelectObjectContent non costituisce un accesso che classifica gli oggetti fino a un livello di accesso frequente. Inoltre, non impedisce la suddivisione degli oggetti dal livello Frequent Access al livello Infrequent Access e quindi al livello Archive Instant Access.

Puoi configurare S3 Intelligent-Tiering come classe di archiviazione predefinita per i dati appena creati specificando `INTELLIGENT-TIERING` nell'intestazione della richiesta PutBucketIntelligentTieringConfiguration. S3 Intelligent-Tiering è progettato per una disponibilità del 99,9% e una durata del 99,999999999%.

> **Note**
>
> Se le dimensioni di un oggetto sono inferiori a 128 KB, questo non è monitorato e il tiering automatico non sarà consentito. Gli oggetti più piccoli vengono sempre archiviati nel livello Accesso frequente.

Livelli di accesso S3 Intelligent-Tiering

Di seguito vengono illustrati i diversi livelli di accesso automatici e facoltativi. Quando gli oggetti vengono spostati tra i livelli di accesso, la classe di archiviazione rimane la stessa (S3 Intelligent-Tiering).

Livello Accesso frequente (automatico)

Questo è il livello di accesso predefinito in cui qualsiasi oggetto creato o trasferito in S3 Intelligent-Tiering inizia il suo ciclo di vita. Un oggetto rimane in questo livello finché viene eseguito l'accesso ad esso. Il livello Frequent Access offre bassa latenza ed elevata velocità di trasmissione effettiva.

Livello Accesso infrequente (automatico)

Se non si accede a un oggetto per 30 giorni consecutivi, l'oggetto passa al livello Accesso infrequente. Il livello Infrequent Access offre bassa latenza ed elevate prestazioni di velocità di trasmissione effettiva.

Livello Archive Instant Access (automatico)

Se non si accede a un oggetto per 90 giorni consecutivi, l'oggetto passa al livello Archive Instant Access. Il livello Archive Instant Access offre bassa latenza ed elevate prestazioni di velocità di trasmissione effettiva.

Livello Accesso di archiviazione (facoltativo)

S3 Intelligent-Tiering offre la possibilità di attivare il livello Archive Access per i dati a cui è possibile accedere in modo asincrono. Dopo l'attivazione, il livello Archive Access archivia automaticamente gli oggetti a cui non è stato eseguito l'accesso per un minimo di 90 giorni consecutivi. Puoi estendere il momento dell'ultimo accesso per l'archiviazione a un massimo di 730 giorni. Il livello Archive Access ha le stesse prestazioni della classe di archiviazione S3 Glacier Flexible Retrieval.

I tempi di recupero standard per questo livello di accesso possono variare dalle 3 alle 5 ore. Se avvii la richiesta di ripristino utilizzando le Operazioni in batch S3, il ripristino inizia in pochi minuti. Per ulteriori informazioni sulle opzioni e gli orari di recupero, consulta the section called “Ripristino degli oggetti dai livelli Archive Access e Deep Archive Access di S3 Intelligent-Tiering”.

> **Note**
>
> Attivare il livello Archive Access per 90 giorni solo se si desidera ignorare il livello Archive Instant Access. Il livello Accesso archivio offre costi di archiviazione leggermente inferiori con tempi di recupero nell'arco di minuti o ore. Il livello Archive Instant Access (Archiviazione con accesso istantaneo) offre un accesso in millisecondi e prestazioni a elevata velocità di trasmissione effettiva.

Livello Accesso di archiviazione profonda (opzionale)

S3 Intelligent-Tiering offre la possibilità di attivare il livello Deep Archive Access per i dati a cui è possibile accedere in modo asincrono. Dopo l'attivazione, il livello Deep Archive Access archivia automaticamente gli oggetti a cui non è stato eseguito l'accesso per un minimo di 180 giorni consecutivi. Puoi estendere il momento dell'ultimo accesso per l'archiviazione a un massimo di 730 giorni. Il livello Accesso di archiviazione profonda ha le stesse prestazioni della classe di archiviazione S3 Glacier Deep Archive.

Il recupero standard degli oggetti in questo livello di accesso avviene entro 12 ore. Se avvii la richiesta di ripristino utilizzando le Operazioni in batch S3, il ripristino inizia nell'arco di 9 ore. Per ulteriori informazioni sulle opzioni e gli orari di recupero, consulta the section called “Ripristino degli oggetti dai livelli Archive Access e Deep Archive Access di S3 Intelligent-Tiering”.

> **Note**
>
> Attiva i livelli Accesso di archiviazione e Accesso di archiviazione profonda solo se l'applicazione può accedere agli oggetti in modo asincrono. Se l'oggetto recuperato è archiviato nei livelli Archive Access o Deep Archive Access, prima devi ripristinarlo utilizzando `RestoreObject`.

Utilizzare S3 Intelligent-Tiering

Puoi utilizzare la classe di archiviazione S3 Intelligent-Tiering per ottimizzare automaticamente i costi di archiviazione. S3 Intelligent-Tiering offre risparmi automatici sui costi spostando i dati a livello granulare degli oggetti tra i livelli di accesso quando i modelli di accesso cambiano. Per i dati a cui è possibile accedere in modo asincrono, puoi scegliere di abilitare l'archiviazione automatica all'interno della classe di archiviazione S3 Intelligent-Tiering utilizzando AWS Management Console, AWS CLI o l'API Amazon S3.

Trasferimento dei dati in S3 Intelligent-Tiering

Sono disponibili due modi per spostare i dati in S3 Intelligent-Tiering. Puoi eseguire direttamente l'operazione PUT dei dati in Piano intelligente Amazon S3 specificando `INTELLIGENT_TIERING` nell'intestazione `x-amz-storage-class` o impostare le configurazioni del ciclo di vita S3 per la transizione di oggetti da S3 Standard o Accesso Infrequente Amazon S3 Standard a Piano intelligente Amazon S3.

Caricamento dei dati in S3 Intelligent-Tiering utilizzando PUT diretto

Quando carichi un oggetto nella classe di archiviazione S3 Intelligent-Tiering utilizzando l'operazione API PUT, specifichi S3 Intelligent-Tiering nell'intestazione della richiesta `x-amz-storage-class`.

La seguente richiesta archivia l'immagine, `my-image.jpg`, nel bucket `myBucket`. La richiesta utilizza l'intestazione `x-amz-storage-class` per richiedere che l'oggetto venga archiviato utilizzando la classe di archiviazione 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
```

Trasferimento dei dati a S3 Intelligent-Tiering da S3 Standard o S3 Standard-Infrequent Access tramite il ciclo di vita S3

Puoi aggiungere regole a una configurazione del ciclo di vita S3 per indicare ad Amazon S3 di trasferire gli oggetti da una classe di archiviazione a un'altra. Per informazioni sulle transizioni supportate e sui vincoli correlati, consulta Trasferimento degli oggetti utilizzando il ciclo di vita S3.

Puoi specificare le configurazioni del ciclo di vita S3 a livello di bucket o di prefisso. In questa regola di configurazione del ciclo di vita S3, il filtro specifica un prefisso della chiave (`documents/`). Pertanto la regola si applica agli oggetti con il prefisso del nome della chiave `documents/`, ad esempio `documents/doc1.txt` e `documents/doc2.txt`. La regola specifica un'azione `Transition` che indica ad Amazon S3 di trasferire gli oggetti alla classe di archiviazione S3 Intelligent-Tiering 0 giorni dopo la creazione. In questo caso, gli oggetti sono idonei per la transizione a S3 Intelligent-Tiering alla mezzanotte UTC successiva alla creazione.

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>
```

Accesso ai livelli S3 Intelligent-Tiering Archive Access e Deep Archive Access

Per ottenere i costi di archiviazione più bassi su dati accessibili in poche ore o minuti, è possibile attivare uno o entrambi i livelli di accesso di archiviazione creando una configurazione a livello di

bucket, prefisso o di tag oggetto utilizzando la AWS Management Console, la AWS CLI o l'API Amazon S3.

Utilizzo della console S3

Per abilitare l'archiviazione automatica di S3 Intelligent-Tiering

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket scegli il nome del bucket desiderato.
3. Scegli Properties (Proprietà).
4. Passa alla sezione S3 Intelligent-Tiering Archive configurations (Configurazioni di archiviazione di S3 Intelligent-Tiering) e scegliCreate configuration (Crea configurazione).
5. Nella sezione Archive configuration settings (Impostazioni di configurazione archivio), specifica un nome descrittivo per la configurazione dell'archivio S3 Intelligent-Tiering.
6. In Choose a configuration scope (Scegli un ambito di configurazione), scegli l'ambito di configurazione da utilizzare. Facoltativamente, puoi limitare l'ambito di configurazione agli oggetti specificati all'interno di un bucket utilizzando un prefisso condiviso, un tag oggetto o una combinazione dei due.
   a. Per limitare l'ambito della configurazione, seleziona Limit the scope of this configuration using one or more filters (Limita l'ambito di questa configurazione utilizzando uno o più filtri).
   b. Per limitare l'ambito della configurazione utilizzando un singolo prefisso, inserisci il prefisso in Prefisso.
   c. Per limitare l'ambito della configurazione utilizzando i tag oggetto, seleziona Add tag (Aggiungi tag) e inserisci un valore per la chiave.
7. In Status (Stato), seleziona Enable (Abilita).
8. Nella sezione Archive settings (Impostazioni archivio), seleziona uno o entrambi i livelli Accesso di archiviazione per abilitarli.
9. Seleziona Crea.

Utilizzo di AWS CLI

Per gestire le configurazioni di S3 Intelligent-Tiering, puoi utilizzare i comandi AWS CLI riportati di seguito:

- delete-bucket-intelligent-tiering-configuration
- get-bucket-intelligent-tiering-configuration
- list-bucket-intelligent-tiering-configurations
- put-bucket-intelligent-tiering-configuration

Per istruzioni sull'impostazione di AWS CLI, consulta Sviluppo con Amazon S3 tramite la AWS CLI.

Quando utilizzi AWS CLI non puoi specificare la configurazione come file XML. È necessario specificare invece il JSON. Di seguito è riportato un esempio di configurazione XML di S3 Intelligent-Tiering e l'equivalente JSON che puoi specificare in un comando AWS CLI.

L'esempio seguente assegna una configurazione S3 Intelligent-Tiering al bucket specificato.

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>
```

Utilizzo dell'operazione API PUT

Puoi utilizzare l'operazione PutBucketIntelligentTieringConfiguration per un bucket specificato e fino a 1.000 configurazioni S3 Intelligent-Tiering per bucket. Puoi definire quali oggetti all'interno di un bucket sono idonei per i livelli di accesso di archiviazione utilizzando un prefisso condiviso o un tag oggetto. Usare un prefisso condiviso o un tag oggetto permette l'allineamento a

determinate applicazioni aziendali, flussi di lavoro, o organizzazioni interne. Hai inoltre la flessibilità necessaria per attivare il livello Accesso di archiviazione, il livello Accesso di archiviazione profonda o entrambi.

Nozioni di base su Piano intelligente S3

Per saperne di più su come usare la classe di archiviazione S3 Intelligent-Tiering (Piano intelligente S3), consulta Tutorial: Guida introduttiva a Piano intelligente Amazon S3.

Gestione di S3 Intelligent-Tiering

La classe di archiviazione S3 Intelligent-Tiering offre risparmi automatici sui costi di archiviazione in tre livelli di accesso a bassa latenza ed elevata velocità di trasmissione effettiva. Inoltre offre funzionalità di archiviazione opzionali che permettono di ottenere costi di archiviazione più bassi nel cloud per i dati accessibili nel giro di minuti o ore. La classe di archiviazione S3 Intelligent-Tiering supporta tutte le funzionalità di Amazon S3, tra cui:

- S3 Inventory, per verificare il livello di accesso degli oggetti
- S3 Replication, per replicare i dati su qualsiasi Regione AWS
- S3 Storage Lens, per visualizzare i parametri di utilizzo e attività dell'archiviazione
- Crittografia lato server, per la protezione dei dati degli oggetti
- S3 Object Lock, per prevenire l'eliminazione accidentale
- AWS PrivateLink, per accedere ad Amazon S3 tramite un endpoint privato in un cloud privato virtuale (VPC)

Identificazione del livello di accesso S3 Intelligent-Tiering in cui sono archiviati gli oggetti

Per ottenere un elenco dei tuoi oggetti e dei relativi metadati corrispondenti, incluso il livello di accesso S3 Intelligent-Tiering, puoi utilizzare. the section called “Gestione dell'inventario” Inventario S3 fornisce CSV, ORC o file di output Parquet che elencano gli oggetti e i metadati corrispondenti. Puoi ricevere questi report di inventario su base giornaliera o settimanale per un bucket Amazon S3 o un prefisso condiviso. (Prefisso condiviso si riferisce agli oggetti con nomi che iniziano con una stringa comune.)

Visualizzazione dello stato dell'archivio di un oggetto all'interno di S3 Intelligent-Tiering

Per ricevere un avviso quando un oggetto all'interno della classe di archiviazione S3 Intelligent-Tiering è passato al livello Accesso archivio o al livello Accesso archivio approfondito, puoi impostare le notifiche di evento Amazon S3. Per ulteriori informazioni, consulta Abilitare le notifiche eventi.

Amazon S3 può pubblicare notifiche di eventi in un argomento Amazon Simple Notification Service (Amazon SNS), una coda Amazon Simple Queue Service (Amazon SQS) o una funzione AWS Lambda . Per ulteriori informazioni, consulta Notifiche di eventi Amazon S3.

Quello che segue è un esempio di un messaggio inviato da Amazon S3 per pubblicare un evento `s3:IntelligentTiering`. Per ulteriori informazioni, consulta the section called “Struttura del messaggio di 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"
            }
      }
   ]
}
```

Puoi utilizzare anche una richiesta di oggetto HEAD per visualizzare lo stato di archiviazione di un oggetto. Se un oggetto viene archiviato utilizzando la classe di archiviazione S3 Intelligent-Tiering e si trova in uno dei livelli di archivio, la richiesta di oggetto HEAD mostrerà il livello di archiviazione corrente. Per mostrare il livello di archiviazione, la richiesta utilizza l'intestazione x-amz-archive-status.

La seguente richiesta di oggetto HEAD restituisce i metadati di un oggetto (in questo 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=
```

Le richieste di oggetto HEAD possono essere utilizzate anche per monitorare lo stato di una richiesta `restore-object`. Se il ripristino dell'archivio è in corso, la richiesta di oggetto HEADincluderà l'intestazione x-amz-restore.

Di seguito è riportato un esempio di risposta di oggetto HEAD che mostra un oggetto archiviato utilizzando S3 Intelligent-Tiering con una richiesta di ripristino in corso.

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
```

Ripristino degli oggetti dai livelli Archive Access e Deep Archive Access di S3 Intelligent-Tiering

Per accedere agli oggetti nei livelli S3 Intelligent-Tiering Archive Access e Deep Archive Access, è necessario avviare una richiesta di ripristino e quindi attendere che l'oggetto venga spostato nel livello Frequent Access. Per ulteriori informazioni sugli oggetti archiviati, consulta. the section called “Utilizzo di oggetti archiviati”

Quando esegui il ripristino dai livelli Accesso di archiviazione o di archiviazione profonda, l'oggetto passa nuovamente al livello Accesso frequente. In seguito, se non accedi all'oggetto per 30 giorni consecutivi, l'oggetto verrà spostato automaticamente nel livello Accesso infrequente. Dopodiché, dopo un minimo di 90 giorni consecutivi senza accesso, l'oggetto passa al livello Accesso archivio. Dopo un minimo di 180 giorni consecutivi senza accesso, l'oggetto passa al livello Accesso archivio approfondito. Per ulteriori informazioni, consulta the section called “Come funziona S3 Intelligent-Tiering”.

Puoi ripristinare un oggetto archiviato utilizzando la console Amazon S3, S3 Batch Operations, l'API REST di Amazon S3, AWS gli SDK o (). AWS Command Line Interface AWS CLI Per ulteriori informazioni, consulta the section called “Utilizzo di oggetti archiviati”.

Gestione del ciclo di vita dello storage

Per gestire i tuoi oggetti in modo che vengano archiviati in modo conveniente per tutto il loro ciclo di vita, crea una configurazione del ciclo di vita di Amazon S3. Una configurazione del ciclo di vita di Amazon S3 è un insieme di regole che definiscono le azioni che Amazon S3 applica a un gruppo di oggetti. Esistono due tipi di operazioni:

- Operazioni di transizione – Queste operazioni definiscono quando gli oggetti passano a un'altra classe di archiviazione. Ad esempio, si può scegliere di trasferire gli oggetti alla classe di archiviazione S3 Standard-IA 30 giorni dopo la creazione o di archiviarli nella classe di archiviazione S3 Glacier Flexible Retrieval un anno dopo la creazione. Per ulteriori informazioni, consulta Utilizzo delle classi di storage di Amazon S3.

  Le richieste di transizione del ciclo di vita sono soggette a costi. Per informazioni sui prezzi, consulta Prezzi di Amazon S3.
- Operazioni di scadenza – Queste operazioni consentono di specificare la scadenza degli oggetti. Gli oggetti scaduti vengono eliminati da Amazon S3 per conto dell'utente.

  I costi di scadenza del ciclo di vita dipendono dalla data di scadenza degli oggetti selezionata. Per ulteriori informazioni, consulta Oggetti in scadenza.

> **Important**
>
> Non puoi utilizzare una bucket policy per impedire eliminazioni o transizioni in base a una regola del ciclo di vita di S3. Ad esempio, anche se la tua bucket policy nega tutte le azioni per tutti i principali, la configurazione di S3 Lifecycle funziona comunque normalmente.

Oggetti esistenti e nuovi

Quando si aggiunge una configurazione del ciclo di vita a un bucket, le regole di configurazione si applicano sia agli oggetti esistenti sia a quelli che vengono aggiunti in un secondo momento. Ad esempio, se oggi aggiungi una regola di configurazione del ciclo di vita con un'azione di scadenza che causa la scadenza degli oggetti 30 giorni dopo la creazione, Amazon S3 metterà in coda per rimuovere tutti gli oggetti esistenti che hanno più di 30 giorni.

Modifiche nella fatturazione

In caso di ritardo tra il momento in cui un oggetto diventa idoneo per un'operazione del ciclo di vita e il momento in cui Amazon S3 trasferisce o rimuove l'oggetto, le modifiche alla fatturazione vengono applicate non appena l'oggetto diventa idoneo per l'operazione del ciclo di vita. Ad esempio, se è prevista la scadenza di un oggetto e Amazon S3 non lo fa scadere immediatamente, non ti verrà addebitato lo spazio di archiviazione dopo l'ora di scadenza.

L'unica eccezione a questo comportamento è se si dispone di una regola del ciclo di vita per il trasferimento alla classe di archiviazione S3 Intelligent-Tiering. In questo caso, le modifiche

alla fatturazione non si verificano fino a quando l'oggetto non è stato trasferito alla classe Piano intelligente Amazon S3.

Per ulteriori informazioni sulle regole del ciclo di vita S3, consulta Elementi della configurazione del ciclo di vita.

Monitoraggio dell'effetto delle regole del ciclo di vita

Per monitorare l'effetto degli aggiornamenti effettuati dalle regole attive del ciclo di vita, vedere. the section called “Come posso monitorare le azioni intraprese dalle mie regole del ciclo di vita?”

Gestione del ciclo di vita degli oggetti

Si consiglia di definire le regole di configurazione del ciclo di vita S3 per oggetti con un ciclo di vita ben definito. Ad esempio:

- Se vengono caricati dei log periodici su un bucket, l'applicazione potrebbe averne bisogno per una settimana o per un mese. Dopo tale periodo, si potrebbe desiderare di eliminarli.
- Alcuni documenti vengono utilizzati con frequenza per un periodo di tempo limitato, Dopo tale momento, vi si accede raramente. Con il passare del tempo, potrebbe non essere più necessario avere l'accesso in tempo reale a questi oggetti. Tuttavia, per motivi legati all'organizzazione o alle normative, potrebbe essere necessario conservarli in archivio per un periodo specifico. Dopo tale periodo, è possibile eliminarli.
- È possibile caricare alcuni tipi di dati su Amazon S3 principalmente per finalità di archiviazione. Ad esempio, è possibile conservare archivi multimediali, record sanitari e finanziari, dati di sequenza genomica non elaborati, backup di database a lungo termine e dati che devono essere conservati per esigenze di conformità normativa.

Tramite le regole di configurazione del ciclo di vita S3, è possibile indicare a Amazon S3 di trasferire gli oggetti in classi di archiviazione meno costose, archiviarli o eliminarli.

Creazione di una configurazione del ciclo di vita

Una configurazione del ciclo di vita S3 è un file XML contenente un set di regole con operazioni predefinite che si desidera che Amazon S3 esegua sugli oggetti durante il loro ciclo di vita.

Puoi creare una configurazione del ciclo di vita utilizzando la console Amazon S3, l'API REST AWS , gli SDK e (). AWS Command Line Interface AWS CLI Per ulteriori informazioni, consulta Impostazione di una configurazione del ciclo di vita su un bucket.

Amazon S3 offre un insieme di operazioni REST API per gestire la configurazione del ciclo di vita in un bucket. Amazon S3 archivia la configurazione come sottorisorsa del ciclo di vita collegata al bucket. Per ulteriori dettagli, consultare la sezione seguente:

- PutBucketLifecycleConfiguration
- GetBucketLifecycleConfiguration
- DeleteBucketLifecycle

Per ulteriori informazioni sulla creazione di una configurazione del ciclo di vita, consulta i seguenti argomenti:

Argomenti

- Trasferimento degli oggetti utilizzando il ciclo di vita Amazon S3
- Oggetti in scadenza
- Impostazione di una configurazione del ciclo di vita su un bucket
- Configurazioni del ciclo di vita e altre configurazioni del bucket
- Configurazione delle notifiche di eventi del ciclo di vita
- Elementi della configurazione del ciclo di vita
- Esempi di configurazione del ciclo di vita S3

Trasferimento degli oggetti utilizzando il ciclo di vita Amazon S3

È possibile aggiungere regole in una configurazione del ciclo di vita per indicare a Amazon S3 di trasferire gli oggetti in un'altra classe di storage di Amazon S3. Per ulteriori informazioni sulle classi di storage, consulta Utilizzo delle classi di storage di Amazon S3. Alcuni esempi di quando è possibile utilizzare le configurazioni del ciclo di vita S3 in questo modo includono i seguenti:

- Quando ci sono oggetti con accesso non frequente, puoi trasferirli nella classe di storage S3 Standard-IA.
- Potresti voler archiviare oggetti a cui non devi accedere in tempo reale nelle classi di storage S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive.

Oggetti esistenti e nuovi

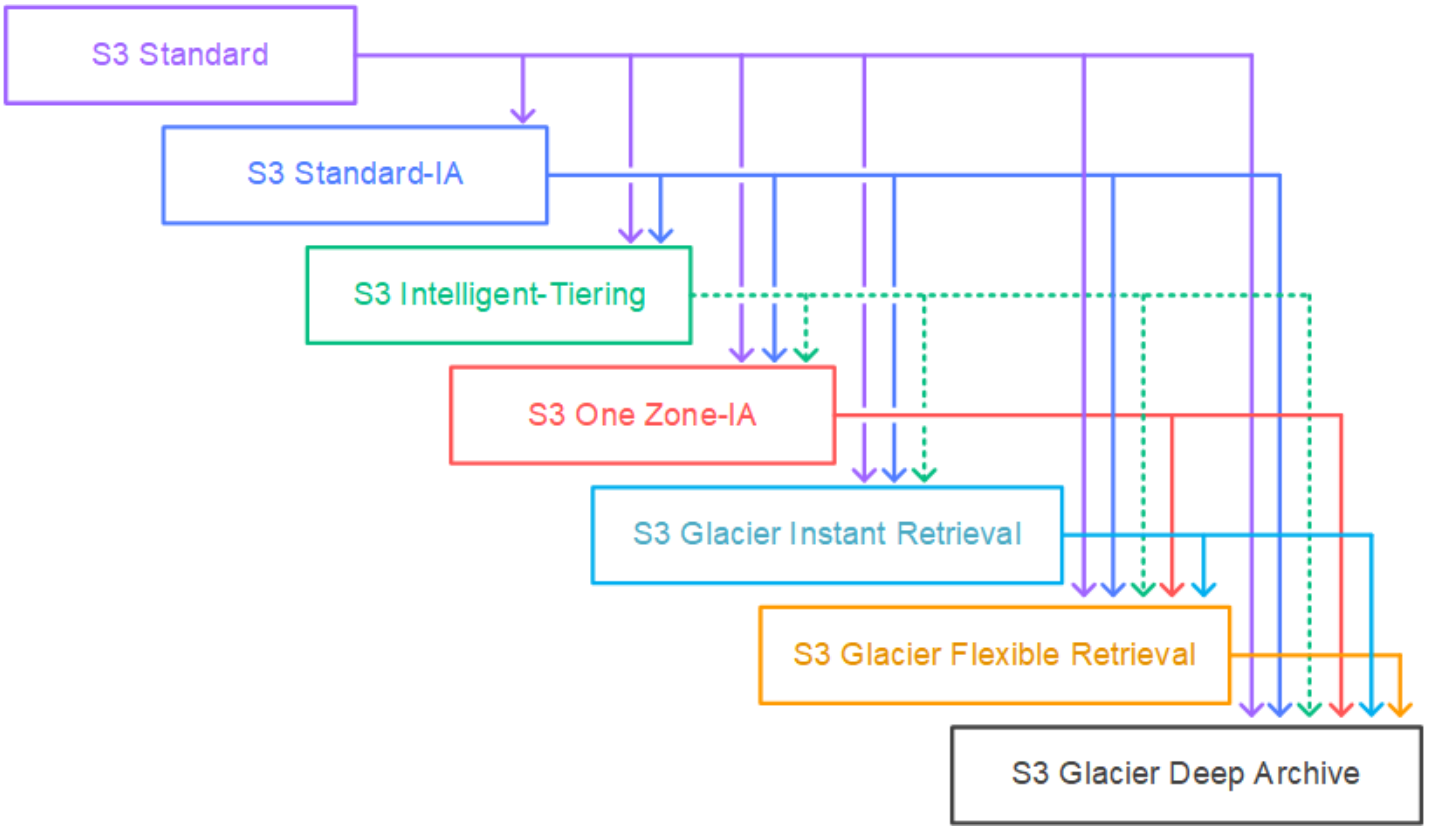
Quando si aggiunge una configurazione del ciclo di vita a un bucket, le regole di configurazione si applicano sia agli oggetti esistenti sia a quelli che vengono aggiunti in un secondo momento. Ad esempio, se oggi aggiungi una regola di configurazione del ciclo di vita con un'azione di transizione che fa sì che gli oggetti con un prefisso specifico passino a una classe di storage diversa 30 giorni dopo la creazione, Amazon S3 metterà in coda per la transizione tutti gli oggetti esistenti che hanno più di 30 giorni e che hanno il prefisso specificato.

> **Important**
>
> Non puoi utilizzare una policy bucket per impedire eliminazioni o transizioni in base a una regola del ciclo di vita S3. Ad esempio, anche se la tua bucket policy nega tutte le azioni per tutti i principali, la configurazione di S3 Lifecycle funziona comunque normalmente.

Trasferimenti supportati e vincoli correlati

In una configurazione del ciclo di vita S3 puoi definire regole per la transizione degli oggetti da una classe di storage a un'altra per risparmiare sui costi di storage. Quando non si conoscono i modelli di accesso degli oggetti o se cambiano nel tempo, è possibile eseguire la transizione degli oggetti alla classe di archiviazione S3 Intelligent-Tiering per ottenere risparmi sui costi in modo automatico. Per informazioni sulle classi di storage, consultare Utilizzo delle classi di storage di Amazon S3.

Amazon S3 supporta un modello a cascata per la transizione tra classi di storage, come mostrato nel diagramma seguente.



Transizioni del ciclo di vita supportate

Amazon S3 supporta le transizioni del ciclo di vita seguenti tra classi di storage tramite una configurazione del ciclo di vita S3.

È possibile effettuare la transizione da:

- La classe di storage S3 Standard a qualsiasi altra classe di storage.
- La classe di archiviazione S3 Standard-IA alle classi S3 Intelligent-Tiering, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive.
- La classe di archiviazione S3 Intelligent-Tiering alle classi di archiviazione S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive.

> **Note**
>
> Esistono alcune eccezioni per la transizione di oggetti dalla classe di storage S3 Intelligent-Tiering a S3 One Zone-IA e ad alcune classi di storage S3 Glacier. Per ulteriori informazioni, consulta the section called “Transizioni del ciclo di vita non supportate”.

- La classe di archiviazione S3 One Zone-IA alle classi di archiviazione S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive.
- La classe di archiviazione S3 Glacier Instant Retrieval alle classi di archiviazione S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive.
- Classe di archiviazione S3 Glacier Flexible Retrieval alla classe S3 Glacier Deep Archive.
- Qualsiasi classe di archiviazione alla classe di archiviazione S3 Glacier Deep Archive.

> **Note**
>
> Non sono previsti costi di recupero dei dati per le transizioni del ciclo di vita. Tuttavia, sono previsti costi di inserimento per richiesta quando si utilizzano regole del ciclo di vita per spostare i dati in PUT qualsiasi COPY classe di storage S3. Considera il costo di inserimento o transizione prima di spostare gli oggetti in qualsiasi classe di storage. Per ulteriori considerazioni relative ai costi, consulta Prezzi di Amazon S3.

Transizioni del ciclo di vita non supportate

Amazon S3 non supporta nessuna delle transizioni del ciclo di vita descritte di seguito.

Non è possibile effettuare la transizione da:

- Qualsiasi classe di storage alla classe di storage S3 Standard.
- Qualsiasi classe di archiviazione alla classe Reduced Redundancy Storage (RRS).
- Classe di archiviazione S3 One Zone-IA alle classi S3 Intelligent-Tiering, S3 Standard-IA o S3 Glacier Instant Retrieval.
- Dalla classe di storage S3 Intelligent-Tiering (tutti i livelli) alla classe di storage S3 Standard-IA.
- La classe di storage S3 Intelligent-Tiering, dal livello Archive Instant Access a S3 One Zone-IA.
- Il livello Archive Access della classe di storage S3 Intelligent-Tiering per S3 One Zone-IA o S3 Glacier Instant Retrieval.
- La classe di storage S3 Intelligent-Tiering Deep Archive Access passa a S3 One Zone-IA, S3 Glacier Instant Retrieval o S3 Glacier Flexible Retrieval.

Vincoli

Alle transizioni tra classi di storage del ciclo di vita si applicano i vincoli seguenti:

Dimensioni degli oggetti e transizioni da S3 Standard o S3 Standard - accesso infrequente a S3 Intelligent-Tiering, S3 Standard-IA o S3 One Zone-IA

Quando esegui una transizione di oggetti dalla classe di archiviazione S3 Standard o S3 Standard-IA - a S3 Intelligent-Tiering, S3 Standard-IA o S3 One Zone-IA, i applicano i vincoli seguenti alle dimensioni degli oggetti:

- Oggetti più grandi – Per le transizioni seguenti vi è un vantaggio in termini di costi per il trasferimento di oggetti più grandi:
  - Dalla classe di storage S3 Standard o S3 Standard-IA a S3 Intelligent-Tiering.
  - Dalla classe di storage S3 Standard a S3 Standard-IA o a S3 One Zone-IA.
- Oggetti di dimensioni inferiori a 128 KiB: per le seguenti transizioni, Amazon S3 non esegue la transizione di oggetti di dimensioni inferiori a 128 KiB:
  - Dalle classi di archiviazione S3 Standard o S3 Standard-IA a S3 Intelligent-Tiering o S3 Glacier Instant Retrieval.
  - Dalla classe di archiviazione S3 Standard a S3 Standard-IA o S3 One Zone-IA.

> **Note**
>
> È possibile filtrare le regole del ciclo di vita in base alle dimensioni dell'oggetto.

> **Important**
>
> Quando sono presenti più regole in una configurazione S3 Lifecycle, un oggetto può diventare idoneo per più azioni S3 Lifecycle nello stesso giorno. In questi casi, Amazon S3 segue le seguenti regole generali:
>
> - L'eliminazione permanente ha la precedenza sul trasferimento.
> - La transizione ha la precedenza sulla creazione di marker di eliminazione.
> - Quando un oggetto è idoneo sia per una transizione S3 Glacier Flexible Retrieval che S3 Standard-IA (o S3 One Zone-IA), Amazon S3 sceglie la transizione S3 Glacier Flexible Retrieval.

Per alcuni esempi, consulta Esempio 5: Sovrapposizione di filtri, conflitto tra operazioni del ciclo di vita e comportamento di Amazon S3 con bucket senza versione.

Giorni minimi per la transizione a S3 Standard-IA o S3 One Zone-IA

Prima di passare gli oggetti a S3 Standard-IA o S3 One Zone-IA, è necessario archiviarli almeno 30 giorni in Amazon S3. Ad esempio, non è possibile creare una regola del ciclo di vita che trasferisca nella classe di storage S3 Standard-IA oggetti creati da un solo giorno. Amazon S3 non supporta questa transizione entro i primi 30 giorni perché gli oggetti più recenti sono spesso accessibili più frequentemente o eliminati prima di quanto sia possibile per lo storage S3 Standard-IA o S3 One Zone-IA.

Analogamente, se si effettua la transizione di oggetti non correnti (in bucket con versione), è possibile passare solo gli oggetti non correnti di almeno 30 giorni a S3 Standard-IA o a S3 One Zone-IA. Per un elenco della durata minima di archiviazione per tutte le classi di archiviazione, vedere. Confronto delle classi di storage di Amazon S3

Addebito minimo di archiviazione di 30 giorni per S3 Standard-IA e S3 One Zone-IA.

Per le classi di archiviazione S3 Standard-IA e S3 One Zone-IA, è previsto un addebito minimo di archiviazione di 30 giorni. Pertanto, non è possibile specificare una singola regola del ciclo di vita sia per una transizione S3 Standard-IA o S3 One Zone-IA, sia per una transizione S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive quando la transizione S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive avviene meno di 30 giorni dopo la transizione S3 Standard-IA o S3 One Zone-IA.

Lo stesso minimo di 30 giorni trova applicazione quando specifichi un trasferimento dall'archiviazione S3 Standard-IA a S3 One Zone-IA. È possibile specificare due regole per ottenere tale risultato, ma occorre pagare per i costi di storage minimi. Per ulteriori considerazioni relative ai costi, consulta Prezzi di Amazon S3.

Gestione del ciclo di vita completo di un oggetto

Per gestire l'intero ciclo di vita S3 di un oggetto, è possibile combinare le operazioni sul ciclo di vita sopra citate. Supponiamo ad esempio di creare oggetti con un ciclo di vita ben definito. Nei primi 30 giorni gli oggetti vengono utilizzati frequentemente. Poi, gli oggetti vengono utilizzati raramente fino a 90 giorni. Dopo tale periodo, gli oggetti non sono più necessari ed è quindi possibile archiviarli o eliminarli.

In questo scenario, è possibile creare una regola del ciclo di vita S3 in cui si specifica l'operazione di transizione iniziale all'archiviazione S3 Intelligent-Tiering, S3 Standard-IA o S3 One Zone-IA, un'altra azione di transizione all'archiviazione Glacier S3 Flexible Retrieval per l'archiviazione e un'operazione di scadenza. Quando si spostano gli oggetti da una classe di archiviazione a un'altra, si risparmia sui costi di archiviazione. Per ulteriori considerazioni relative ai costi, consulta Prezzi di Amazon S3.

Trasferimento nelle classi di archiviazione S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive (archiviazione di oggetti)

Utilizzando una configurazione S3 Lifecycle, puoi trasferire gli oggetti alle classi di storage S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive per l'archiviazione. Quando scegli le classi di archiviazione S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, gli oggetti rimangono in Amazon S3. Non puoi accedervi direttamente tramite il servizio Amazon S3 Glacier separato. Per informazioni più generali su S3 Glacier consulta Che cos'è Amazon S3 Glacier nella Guida per Developer di Amazon S3 Glacier.

Prima di archiviare gli oggetti, consultare le sezioni seguenti per alcune considerazioni in merito.

Considerazioni generali

Di seguito sono riportate le considerazioni generali di cui tenere conto prima di archiviare gli oggetti:

- Gli oggetti crittografati rimangono tali durante l'intero processo di transazione tra classi di storage.
- Gli oggetti conservati nelle classi di archiviazione S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive non sono disponibili in tempo reale.

  Gli oggetti archiviati sono oggetti di Amazon S3 , tuttavia, prima di poter accedere a un oggetto archiviato, è necessario ripristinarne una copia temporanea. La copia dell'oggetto ripristinato è disponibile solo per la durata specificata nella richiesta di ripristino. Successivamente Amazon S3 elimina la copia temporanea e l'oggetto rimane archiviato in S3 Glacier Flexible Retrieval.

  Puoi ripristinare un oggetto utilizzando la console Amazon S3 o a livello di codice utilizzando le librerie wrapper AWS SDK o l'API REST di Amazon S3 nel codice. Per ulteriori informazioni, consulta Ripristino di un oggetto archiviato.
- Gli oggetti conservati nella classe di archiviazione S3 Glacier Flexible Retrieval possono essere trasferiti solo nella classe di archiviazione S3 Glacier Deep Archive.

  È possibile utilizzare una regola di configurazione del ciclo di vita S3 per convertire la classe di archiviazione di un oggetto da S3 Glacier Flexible Retrieval alla sola classe di archiviazione S3

Glacier Deep Archive. Se si vuole modificare la classe di archiviazione di un oggetto conservato in Glacier S3 Flexible Retrieval in una classe di archiviazione diversa da S3 Glacier Deep Archive, bisogna prima utilizzare l'operazione di ripristino per creare una copia temporanea dell'oggetto. Usa quindi l'operazione di copia per sovrascrivere l'oggetto specificando S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 One Zone-IA o Reduced Redundancy come classe di storage.

- La transizione di oggetti nella classe di storage S3 Glacier Deep Archive è unidirezionale.

  Non puoi usare una regola di configurazione del ciclo di vita S3 per convertire la classe di storage di un oggetto da S3 Glacier Deep Archive in un'altra classe di storage. Se vuoi cambiare la classe di storage di un oggetto archiviato specificandone un'altra, devi prima di tutto usare l'operazione di ripristino per creare una copia temporanea dell'oggetto. Utilizzare quindi l'operazione di copia per sovrascrivere l'oggetto specificando S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval o Reduced Redundancy Storage come classe di archiviazione.

  > **Note**
  >
  > L'operazione di copia per gli oggetti ripristinati non è supportata nella console Amazon S3 per oggetti nelle classi di storage Recupero flessibile Amazon S3 Glacier o S3 Glacier Deep Archive. Per questo tipo di operazione di copia, usa AWS Command Line Interface (AWS CLI), gli SDK o l'API REST. AWS

  Gli oggetti conservati nelle classi di archiviazione S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive sono visibili e disponibili solo tramite Amazon S3. Non sono disponibili tramite il servizio Amazon S3 Glacier separato.

  Questi sono oggetti Amazon S3 e puoi accedervi solo utilizzando la console Amazon S3 o l'API Amazon S3. Non è possibile accedere agli oggetti archiviati tramite la console Glacier di Amazon S3 separata o l'API di Amazon S3 Glacier.

Considerazioni sui costi

Se si intendono archiviare dati con accesso non frequente per un periodo di mesi o anni, le classi di archiviazione S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive consentono di ridurre i costi di archiviazione. Tuttavia, è consigliabile considerare quanto segue per assicurarsi che la classe di archiviazione S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive sia la scelta appropriata:

- Costi generali di archiviazione – Quando esegui la transizione di oggetti alla classe di archiviazione S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, una quantità fissa di spazio di archiviazione viene aggiunta a ogni oggetto per poter conservare i metadati per la gestione dell'oggetto.
  - Per ogni oggetto archiviato in S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, Amazon S3 utilizzerà 8 KB di spazio di archiviazione per il nome dell'oggetto e gli altri metadati. Amazon S3 memorizza questi metadati per consentire di generare tramite l'API Amazon S3 un elenco in tempo reale degli oggetti archiviati. Per ulteriori informazioni, consulta Get Bucket (List Objects). Questo spazio di archiviazione aggiuntivo viene addebitato secondo le tariffe di S3 Standard.
  - Per ogni oggetto archiviato in S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, Amazon S3 aggiungerà 32 KB di spazio di archiviazione per l'indice e i metadati correlati. Questi dati aggiuntivi sono necessari per identificare e ripristinare l'oggetto desiderato. Questo spazio di archiviazione aggiuntivo vene addebitato secondo le tariffe di S3 Glacier Flexible Retrieval o di S3 Glacier Deep Archive.

  Per l'archiviazione di oggetti di piccole dimensioni, tenere presenti questi costi di storage. Per ridurre i costi aggiuntivi, si possono aggregare diversi oggetti di piccole dimensioni in un numero più contenuto di oggetti di grandi dimensioni.
- Numero di giorni pianificati per conservare gli oggetti archiviati – S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive sono soluzioni di archiviazione a lungo termine. Il periodo minimo di archiviazione è di 90 giorni per la classe di archiviazione S3 Glacier Flexible Retrieval e 180 giorni per S3 Glacier Deep Archive. L'eliminazione dei dati archiviati in Amazon S3 Glacier è gratuita se gli oggetti eliminati sono archiviati per un tempo superiore al periodo di archiviazione minimo. Se elimini o sovrascrivi un oggetto archiviato entro il periodo di durata minimo, Amazon S3 addebita una tariffa ripartita proporzionalmente di eliminazione anticipata. Per informazioni sulla tariffa di eliminazione anticipata, consulta "Come viene addebitata l'eliminazione di oggetti da Amazon S3 Glacier archiviati da meno di 90 giorni?". nelle Domande frequenti su Amazon S3.
- Costi della richiesta di transizione a S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive – Ogni oggetto che viene trasferito nelle classi di archiviazione S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive costituisce una richiesta di transizione. È previsto un costo per ognuna di queste richieste. Se si ha intenzione di trasferire un elevato numero di oggetti, i costi di richiesta vanno tenuti in considerazione. Se stai archiviando una combinazione di oggetti che include oggetti di piccole dimensioni, in particolare quelli di dimensioni inferiori a 128 KB, ti consigliamo di utilizzare il filtro delle dimensioni degli oggetti del ciclo di vita per escludere oggetti di piccole dimensioni dalla transizione per ridurre i costi di richiesta. S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive non bloccano automaticamente la transizione di oggetti di dimensioni inferiori a 128 KB.

- Costi di ripristino dei dati da S3 Glacier e S3 Glacier Deep Archive – S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive sono progettati per l'archiviazione a lungo termine di dati ad accesso non frequente. Per informazioni sulle spese di ripristino dei dati, consulta "Quanto costa recuperare i dati da Amazon S3 Glacier?". nelle Domande frequenti su Amazon S3. Per informazioni su come ripristinare i dati da Amazon S3 Glacier, consulta Ripristino di un oggetto archiviato.

Quando si archiviano oggetti su Amazon S3 Glacier utilizzando la gestione del ciclo di vita S3, Amazon S3 trasferisce questi oggetti in modo asincrono. Potrebbe esserci un ritardo tra la data di trasferimento nella regola di configurazione del ciclo di vita S3 e la data del trasferimento fisico. I costi di Amazon S3 Glacier vengono addebitati in base alla data di transizione specificata nella regola. Per maggiori informazioni, consulta la sezione relativa ad Amazon S3 Glacier nelle domande frequenti su Amazon S3.

Nella pagina dei dettagli del prodotto Amazon S3 sono disponibili informazioni sui prezzi ed esempi di calcolo per l'archiviazione di oggetti di Amazon S3. Per ulteriori informazioni, consultare i seguenti argomenti:

- "Come vengono calcolati i costi di storage per gli oggetti di Amazon S3 archiviati in Amazon S3 Glacier?" nelle Domande frequenti su Amazon S3.
- "Quali costi vengono addebitati per l'eliminazione di oggetti da Amazon S3 Glacier archiviati da meno di 90 giorni?" nelle Domande frequenti su Amazon S3.
- "Quanto costa recuperare dati da Amazon S3 Glacier?" nelle Domande frequenti su Amazon S3.
- Prezzi di Amazon S3 per informazioni sui costi di storage per le diverse classi di storage.

Ripristino di oggetti archiviati

Gli oggetti archiviati non sono accessibili in tempo reale. È necessario innanzitutto avviare una richiesta di ripristino e successivamente attendere che venga resa disponibile una copia temporanea dell'oggetto per la durata specificata nella richiesta. Anche dopo avere ricevuto una copia temporanea dell'oggetto ripristinato, la classe di archiviazione dell'oggetto rimane S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive. (Una richiesta operativa HeadObjecto GetObjectAPI restituirà S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive come classe di storage.)

> **Note**
>
> Quando ripristini un archivio, paghi sia per l'archivio (tariffa per le classi di archiviazione S3 Glacier Flexible Retrieval [Recupero flessibile S3 Glacier] e S3 Glacier Deep Archive

[Archiviazione profonda S3 Glacier]) che per una copia temporaneamente ripristinata (tariffa di archiviazione S3 Standard). Per informazioni sui prezzi, consulta Prezzi di Amazon S3.

Puoi ripristinare una copia dell'oggetto a livello di programmazione oppure utilizzando la console Amazon S3. Amazon S3 elabora una sola richiesta di ripristino alla volta per oggetto. Per ulteriori informazioni, consulta Ripristino di un oggetto archiviato.

Oggetti in scadenza

Quando un oggetto raggiunge la fine del suo ciclo di vita in base alla configurazione del ciclo di vita, Amazon S3 esegue un'azione in base allo stato di S3 Versioning in cui si trova il bucket.

- Bucket senza versione: Amazon S3 mette in coda l'oggetto per la rimozione e lo rimuove in modo asincrono, rimuovendo definitivamente l'oggetto.
- Bucket con controllo delle versioni abilitata: se la versione dell'oggetto corrente non è un contrassegno di eliminazione, Amazon S3 aggiunge un contrassegno di eliminazione con un ID versione univoco. La versione corrente diventa quindi non corrente e il contrassegno di eliminazione diventa la versione corrente.
- Bucket con controllo delle versioni sospesa: Amazon S3 crea un contrassegno di eliminazione il cui ID versione è null. Il contrassegno di eliminazione sostituisce qualsiasi versione dell'oggetto con ID versione null nella gerarchia delle versioni: questa operazione elimina di fatto l'oggetto.

Per un bucket con versione (ovvero, con controllo delle versioni attivata o sospesa), sono diversi i fattori che governano la gestione dell'operazione da parte di Amazon S3. Per i bucket con controllo delle versioni abilitato o sospeso, vale quanto segue:

- La scadenza dell'oggetto si applica solo alla sua versione corrente (non ha effetto sulle versioni non correnti dell'oggetto).
- Amazon S3 non esegue alcuna operazione se sono presenti una o più versioni dell'oggetto e il contrassegno di eliminazione è la versione corrente.
- Se la versione corrente dell'oggetto è l'unica disponibile e porta anche il contrassegno di eliminazione (noto anche come contrassegno di eliminazione dell'oggetto scaduto, dove tutte le versioni degli oggetti vengono eliminate e rimane solo un contrassegno di eliminazione), Amazon S3 rimuove il contrassegno di eliminazione dall'oggetto scaduto. È possibile inoltre utilizzare l'operazione di eliminazione per indicare ad Amazon S3 di rimuovere i contrassegni di

eliminazione dell'oggetto scaduto. Per un esempio, consulta Esempio 7: rimozione dei contrassegni di eliminazione oggetto scaduto.

- Puoi utilizzare l'elemento `NoncurrentVersionExpiration` action per indicare ad Amazon S3 di eliminare definitivamente le versioni non correnti degli oggetti. Questi oggetti eliminati non possono essere recuperati. È possibile basare questa scadenza su un determinato numero di giorni trascorsi da quando gli oggetti non sono più correnti. Oltre al numero di giorni, puoi anche fornire un numero massimo di versioni non correnti da conservare (compreso tra 1 e 100). Questo valore specifica quante versioni non correnti più recenti devono esistere prima che Amazon S3 possa eseguire l'operazione associata su una determinata versione. Per specificare il numero massimo di versioni non correnti, è necessario fornire anche un `Filter` elemento. Se non specifichi un `Filter` elemento, Amazon S3 genera un `InvalidRequest` errore quando fornisci un numero massimo di versioni non correnti. Per ulteriori informazioni sull'utilizzo dell'elemento `NoncurrentVersionExpiration` action, consulta. the section called “Elementi per la descrizione delle operazioni nel ciclo di vita”

Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

> **Important**
>
> Quando sono presenti più regole in una configurazione del ciclo di vita di S3, un oggetto può diventare idoneo per più azioni del ciclo di vita di S3 nello stesso giorno. In questi casi, Amazon S3 segue le seguenti regole generali:
>
> - L'eliminazione permanente ha la precedenza sul trasferimento.
> - La transizione ha la precedenza sulla creazione di marker di eliminazione.
> - Quando un oggetto è idoneo sia per una transizione S3 Glacier Flexible Retrieval che per una transizione S3 Standard-IA (o S3 One Zone-IA), Amazon S3 sceglie la transizione S3 Glacier Flexible Retrieval.
>
> Per alcuni esempi, consulta Esempio 5: Sovrapposizione di filtri, conflitto tra operazioni del ciclo di vita e comportamento di Amazon S3 con bucket senza versione.

Oggetti esistenti e nuovi

Quando si aggiunge una configurazione del ciclo di vita a un bucket, le regole di configurazione si applicano sia agli oggetti esistenti sia a quelli che vengono aggiunti in un secondo momento. Ad

esempio, se oggi aggiungi una regola di configurazione del ciclo di vita con un'azione di scadenza che fa scadere gli oggetti con un prefisso specifico 30 giorni dopo la creazione, Amazon S3 metterà in coda per la rimozione tutti gli oggetti esistenti che hanno più di 30 giorni e che hanno il prefisso specificato.

> **Important**
>
> Non puoi utilizzare una policy bucket per impedire eliminazioni o transizioni in base a una regola S3 Lifecycle. Ad esempio, anche se la tua bucket policy nega tutte le azioni per tutti i principali, la configurazione di S3 Lifecycle funziona comunque normalmente.

Come individuare la data di scadenza degli oggetti

Per scoprire quando è prevista la scadenza di un oggetto, utilizza l'operazione o API. HeadObjectGetObject Queste operazioni API restituiscono intestazioni di risposta che forniscono la data e l'ora in cui l'oggetto non è più inseribile nella cache.

> **Note**
>
> - La data di scadenza dell'oggetto e la data in cui Amazon S3 rimuove un oggetto potrebbero non coincidere. Non è previsto alcun addebito per la scadenza o il tempo di archiviazione associato a un oggetto scaduto.
> - Prima di aggiornare, disabilitare o eliminare le regole del ciclo di vita, utilizza le operazioni `LIST` API (ad esempio ListObjectsV2ListObjectVersions, and ListMultipartUploads) o verifica che Amazon S3 abbia oggetti idonei transitati e scaduti in base Amazon S3 Inventory ai tuoi casi d'uso.

Costo della durata di archiviazione minima

Se crei una regola di scadenza del ciclo di vita S3 che specifica la scadenza di un oggetto presente nell'archiviazione S3 Standard-IA o S3 One Zone-IA per meno di 30 giorni, ti verranno addebitati comunque i costi per i 30 giorni. Se viene creata una regola di scadenza del ciclo di vita che determina la scadenza di oggetti che sono stati nella classe di archiviazione S3 Glacier Flexible Retrieval per meno di 90 giorni, verranno comunque addebitati i costi per 90 giorni. Se viene creata una regola di scadenza del ciclo di vita che determina la scadenza di oggetti che sono stati nella

classe di storage S3 Glacier Deep Archive per meno di 180 giorni, verranno comunque addebitati i costi per 180 giorni.

Per ulteriori informazioni, consulta i Prezzi di Amazon S3.

Impostazione di una configurazione del ciclo di vita su un bucket

Questa sezione spiega come impostare una configurazione del ciclo di vita di Amazon S3 su un bucket utilizzando la console Amazon S3, il AWS Command Line Interface (AWS CLI), gli SDK AWS o l'API REST di Amazon S3. Per informazioni sulla configurazione del ciclo di vita S3, consulta Gestione del ciclo di vita dello storage.

Puoi utilizzare regole del ciclo di vita per definire le operazioni che desideri vengano eseguite da Amazon S3 durante il ciclo di vita di un oggetto, ad esempio la transizione di oggetti in un'altra classe di storage, la relativa archiviazione o l'eliminazione dopo un periodo di tempo specificato.

Prima di impostare una configurazione del ciclo di vita, tieni presente quanto segue:

Ritardo di propagazione della configurazione del ciclo di vita

Quando si aggiunge una configurazione del ciclo di vita S3 in un bucket, la propagazione della configurazione nuova o aggiornata a tutti i sistemi Amazon S3 viene in genere completata con un leggero ritardo. Occorrono alcuni minuti prima che la configurazione venga applicata completamente. Questo ritardo può verificarsi anche quando si elimina una configurazione del ciclo di vita S3.

Ritardo di transizione o scadenza

Esiste un ritardo tra il momento in cui una regola del ciclo di vita viene soddisfatta e il completamento dell'azione relativa alla regola. Ad esempio, supponiamo che un set di oggetti sia scaduto a causa di una regola del ciclo di vita il 1° gennaio. Anche se la regola di scadenza è stata soddisfatta il 1° gennaio, Amazon S3 potrebbe effettivamente eliminare questi oggetti solo giorni o addirittura settimane dopo. Questo ritardo si verifica perché S3 Lifecycle mette in coda gli oggetti per le transizioni o le scadenze in modo asincrono. Tuttavia, le modifiche alla fatturazione vengono generalmente applicate quando la regola del ciclo di vita è soddisfatta, anche se l'azione non è completa. Per ulteriori informazioni, consulta Modifiche alla fatturazione. Per monitorare l'effetto degli aggiornamenti apportati dalle regole attive del ciclo di vita, consulta the section called “Come posso monitorare le azioni intraprese dalle mie regole del ciclo di vita?”

Disattivazione o eliminazione delle regole del ciclo di vita

Quando disabiliti o elimini le regole del ciclo di vita, Amazon S3 interrompe la pianificazione di nuovi oggetti per l'eliminazione o la transizione dopo un piccolo ritardo. Le pianificazioni degli oggetti già pianificati vengono annullate e gli oggetti non vengono eliminati né spostati.

> **Note**
>
> Prima di aggiornare, disabilitare o eliminare le regole del ciclo di vita, utilizza le operazioni `LIST` API (ad esempio ListObjectsV2ListObjectVersions, and ListMultipartUploads) o verifica che Amazon S3 abbia oggetti idonei transitati e scaduti in base Amazon S3 Inventory ai tuoi casi d'uso. Se riscontri problemi con l'aggiornamento, la disabilitazione o l'eliminazione delle regole del ciclo di vita, consulta. Risoluzione dei problemi del ciclo di vita di Amazon S3

Oggetti esistenti e nuovi

Quando si aggiunge una configurazione del ciclo di vita a un bucket, le regole di configurazione si applicano sia agli oggetti esistenti sia a quelli che vengono aggiunti in un secondo momento. Ad esempio, se oggi aggiungi una regola di configurazione del ciclo di vita con un'azione di scadenza che fa scadere gli oggetti con un prefisso specifico 30 giorni dopo la creazione, Amazon S3 metterà in coda per la rimozione tutti gli oggetti esistenti che hanno più di 30 giorni e che hanno il prefisso specificato.

Monitoraggio dell'effetto delle regole del ciclo di vita

Per monitorare l'effetto degli aggiornamenti apportati dalle regole attive del ciclo di vita, vedere the section called “Come posso monitorare le azioni intraprese dalle mie regole del ciclo di vita?”

Modifiche alla fatturazione

Potrebbe verificarsi un ritardo tra il momento in cui le regole di configurazione del ciclo di vita vengono soddisfatte e il momento in cui viene intrapresa l'azione innescata dal rispetto della regola. Tuttavia, le modifiche alla fatturazione avvengono non appena la regola di configurazione del ciclo di vita viene soddisfatta, anche se l'azione non è ancora stata intrapresa.

Ad esempio, dopo la scadenza dell'oggetto, non ti viene addebitato alcun costo per l'archiviazione, anche se l'oggetto non viene eliminato immediatamente. Allo stesso modo, non appena scade il tempo di transizione dell'oggetto, ti vengono addebitate le tariffe di archiviazione di S3 Glacier Flexible Retrieval, anche se l'oggetto non viene immediatamente trasferito alla classe di storage S3 Glacier Flexible Retrieval.

Tuttavia, le transizioni del ciclo di vita alla classe di storage S3 Intelligent-Tiering sono un'eccezione. Le modifiche alla fatturazione avvengono solo dopo la transizione dell'oggetto nella classe di storage S3 Intelligent-Tiering.

Regole multiple o in conflitto

Quando sono presenti più regole in una configurazione S3 Lifecycle, un oggetto può diventare idoneo per più azioni S3 Lifecycle nello stesso giorno. In questi casi, Amazon S3 segue le seguenti regole generali:

- L'eliminazione permanente ha la precedenza sul trasferimento.
- La transizione ha la precedenza sulla creazione di marker di eliminazione.
- Quando un oggetto è idoneo sia per una transizione S3 Glacier Flexible Retrieval che per una transizione S3 Standard-IA (o S3 One Zone-IA), Amazon S3 sceglie la transizione S3 Glacier Flexible Retrieval.

Per alcuni esempi, consulta Esempio 5: Sovrapposizione di filtri, conflitto tra operazioni del ciclo di vita e comportamento di Amazon S3 con bucket senza versione.

Utilizzo della console S3

Puoi definire le regole del ciclo di vita per tutti gli oggetti o per un sottoinsieme di oggetti in un bucket utilizzando un prefisso condiviso (nomi di oggetti che iniziano con una stringa comune) o un tag. Nella regola del ciclo di vita, è possibile definire azioni specifiche per le versioni correnti e non correnti degli oggetti. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Gestione del ciclo di vita dello storage
- Utilizzo della funzione Controllo delle versioni nei bucket S3

Per creare una regola del ciclo di vita

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegliere il nome del bucket per il quale si desidera creare una regola del ciclo di vita.
3. Scegliere la scheda Management (Gestione), quindi Create lifecycle rule (Crea regola ciclo di vita).

4. In Lifecycle rule name (Nome regola ciclo di vita) immettere un nome per la regola.

   Il nome deve essere univoco all'interno del bucket.

5. Scegliere l'ambito della regola del ciclo di vita:

   - Per applicare questa regola del ciclo di vita a tutti gli oggetti con un prefisso o un tag specifico, scegliere Limita l'ambito a prefissi o tag specifici.
     - Per limitare l'ambito in base al prefisso, in Prefix (Prefisso) immettere il prefisso.
     - Per limitare l'ambito in base al tag, scegliere Add tag (Aggiungi tag) e immettere la chiave e il valore del tag.

     Per ulteriori informazioni sui prefissi dei nomi oggetto, consulta Creazione di nomi di chiavi oggetto. Per ulteriori informazioni sui tag degli oggetti, consulta Suddivisione in categorie dello storage utilizzando i tag.
   - Per applicare questa regola del ciclo di vita a tutti gli oggetti nel bucket, scegli Questa regola si applica a tutti gli oggetti nel bucket, quindi scegli Riconosco che questa regola si applica a tutti gli oggetti nel bucket.

6. Per filtrare una regola in base alla dimensione dell'oggetto, puoi selezionare Specificare la dimensione minima dell'oggetto, Specificare la dimensione massima dell'oggetto o entrambe le opzioni.

   - Quando si specifica un valore per Dimensione minima dell'oggetto o Dimensione massima dell'oggetto, il valore deve essere maggiore di 0 byte e massimo 5 TB. È possibile specificare questo valore in byte, KB, MB o GB.
   - Quando specificate entrambi i valori, la dimensione massima dell'oggetto deve essere maggiore della dimensione minima dell'oggetto.

   > **Note**
   >
   > I filtri Dimensione minima dell'oggetto e Dimensione massima dell'oggetto escludono i valori specificati. Ad esempio, se si imposta un filtro per far scadere gli oggetti con una dimensione minima dell'oggetto di 128 KB, gli oggetti che pesano esattamente 128 KB non scadono. La regola si applica invece solo agli oggetti di dimensioni superiori a 128 KB.

7. In Lifecycle rule actions (Operazioni regola ciclo di vita) scegliere le operazioni da far eseguire alla regola del ciclo di vita:

- Transizione delle versioni correnti degli oggetti tra classi di storage
- Transizione delle versioni precedenti degli oggetti tra classi di storage
- Definizione della scadenza delle versioni correnti degli oggetti

> **Note**
>
> Per i bucket che non hanno S3 Versioning abilitato, la scadenza delle versioni correnti fa sì che Amazon S3 elimini definitivamente gli oggetti. Per ulteriori informazioni, consulta the section called “Operazioni del ciclo di vita e stato della funzione Controllo delle versioni nel bucket”.

- Eliminazione permanente delle versioni precedenti degli oggetti
- Eliminazione dei contrassegni di eliminazione o di caricamenti in più parti incompleti

A seconda delle operazioni scelte, vengono visualizzate opzioni diverse.

8. Per eseguire la transizione delle versioni correnti degli oggetti tra classi di storage, in Transition current versions of objects between storage classes (Transizione delle versioni correnti degli oggetti tra classi di storage):

   a. In Storage class transitions, scegli la classe di storage verso cui passare. Per un elenco delle transizioni possibili, consulta. the section called “Transizioni del ciclo di vita supportate” È possibile scegliere tra le seguenti classi di archiviazione:

      - S3 Standard-IA
      - S3 Intelligent-Tiering
      - S3 One Zone-IA
      - S3 Glacier Flexible Retrieval
      - S3 Glacier Deep Archive

   b. In Days after object creation (Giorni dopo la creazione dell'oggetto) immettere il numero di giorni successivi alla creazione dopo i quali eseguire la transizione dell'oggetto.

   Per ulteriori informazioni sulle classi di storage, consulta Utilizzo delle classi di storage di Amazon S3. È possibile definire le transizioni per la versione degli oggetti corrente o per quella precedente, oppure per entrambe le versioni: corrente e precedente. La funzione Controllo

delle versioni consente di gestire più versioni di un oggetto in un unico bucket. Per ulteriori informazioni sulla funzione Controllo delle versioni, consulta Utilizzo della console S3.

> **Important**
>
> Quando scegli le classi di archiviazione S3 Glacier Flexible Retrieval o Glacier Deep Archive, gli oggetti rimangono in Amazon S3. Non puoi accedervi direttamente tramite il servizio Amazon S3 Glacier separato. Per ulteriori informazioni, consulta Trasferimento degli oggetti utilizzando il ciclo di vita Amazon S3.

9. Per trasferire versioni non correnti di oggetti tra classi di archiviazione, in Transizione di versioni non correnti di oggetti tra classi di archiviazione:

   a. In Transizioni delle classi di archiviazione, scegli la classe di archiviazione a cui passare. Per un elenco delle transizioni possibili, consulta. the section called “Transizioni del ciclo di vita supportate” È possibile scegliere tra le seguenti classi di archiviazione:

      - S3 Standard-IA
      - S3 Intelligent-Tiering
      - S3 One Zone-IA
      - S3 Glacier Flexible Retrieval
      - S3 Glacier Deep Archive

   b. In Giorni dopo che l'oggetto diventa non corrente, immettete il numero di giorni dopo la creazione per la transizione dell'oggetto.

10. Per definire la scadenza delle versioni correnti degli oggetti, in Expire current versions of objects (Scadenza versioni attuali degli oggetti) immettere un numero di giorni in Number of days after object creation (Numero di giorni dopo la creazione dell'oggetto).

> **Important**
>
> In un bucket senza versione, l'azione di scadenza comporta la rimozione permanente dell'oggetto da parte di Amazon S3. Per ulteriori informazioni sulle operazioni del ciclo di vita, consulta Elementi per la descrizione delle operazioni nel ciclo di vita.

11. Per eliminare in modo definitivo le versioni precedenti degli oggetti, in Permanently delete noncurrent versions of objects (Elimina in modo definitivo le versioni non aggiornate degli oggetti), specifica il numero di giorni nel campo Days after objects become noncurrent (Numero

di giorni dopo i quali gli oggetti non sono più aggiornati). Facoltativamente puoi specificare il numero di versioni più recenti da mantenere immettendo un valore nel campo Number of newer versions to retain (Numero di versioni più recenti da mantenere).

12. In Delete expired delete markers or incomplete multipart uploads (Elimina contrassegni di eliminazione scaduti o caricamenti in più parti incompleti) scegliere Delete expired object delete markers (Elimina contrassegni di eliminazione oggetti scaduti) e Delete incomplete multipart uploads (Elimina caricamenti in più parti incompleti). Immettere quindi il numero di giorni dopo l'avvio dei caricamenti in più parti dopo i quali si desidera terminare e pulire i caricamenti in più parti incompleti.

    Per ulteriori informazioni sui caricamenti in più parti, consultare Caricamento e copia di oggetti utilizzando il caricamento in più parti.

13. Scegli Crea regola.

    Se la regola non contiene errori, Amazon S3 la abilita ed è possibile visualizzarla nella scheda Management (Gestione) in Lifecycle rules (Regole ciclo di vita).

Per informazioni sui AWS CloudFormation modelli ed esempi, consulta Lavorare con i AWS CloudFormation modelli e AWS::S3::Bucketnella Guida per l'utente.AWS CloudFormation

Utilizzo del AWS CLI

È possibile utilizzare i seguenti AWS CLI comandi per gestire le configurazioni del ciclo di vita di S3:

- `put-bucket-lifecycle-configuration`
- `get-bucket-lifecycle-configuration`
- `delete-bucket-lifecycle`

Per istruzioni sulla configurazione di, consulta. AWS CLISviluppo con Amazon S3 tramite la AWS CLI

La configurazione del ciclo di vita di Amazon S3 è un file XML. Tuttavia, quando si utilizza il AWS CLI, non è possibile specificare il formato XML. È invece necessario specificare il formato JSON. Di seguito sono riportati esempi di configurazioni del ciclo di vita XML e le configurazioni JSON equivalenti che è possibile specificare in un comando. AWS CLI

Prendiamo in considerazione la configurazione del ciclo di vita S3 di esempio riportata di seguito:

Example Esempio 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 Esempio 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
            }
        }
    ]
}
```

È possibile testare il codice `put-bucket-lifecycle-configuration` nel modo seguente:

Per testare la configurazione

1. Salvate la configurazione del ciclo di vita JSON in un file (ad esempio,). *`lifecycle.json`*
2. Esegui il AWS CLI comando seguente per impostare la configurazione del ciclo di vita nel tuo bucket. Sostituire *`user input placeholders`* con le proprie informazioni.

   ```
   $ aws s3api put-bucket-lifecycle-configuration  \
   --bucket DOC-EXAMPLE-BUCKET  \
   --lifecycle-configuration file://lifecycle.json
   ```

3. Per verificare, recupera la configurazione del ciclo di vita di S3 utilizzando il comando seguente: `get-bucket-lifecycle-configuration` AWS CLI

   ```
   $ aws s3api get-bucket-lifecycle-configuration  \
   --bucket DOC-EXAMPLE-BUCKET
   ```

4. Per eliminare la configurazione di S3 Lifecycle, usa il comando come segue: `delete-bucket-lifecycle` AWS CLI

```
aws s3api delete-bucket-lifecycle \
--bucket DOC-EXAMPLE-BUCKET
```

Utilizzo degli SDK AWS

Java

Puoi utilizzare il AWS SDK for Java per gestire la configurazione del ciclo di vita S3 di un bucket. Per ulteriori informazioni sulla gestione della configurazione del ciclo di vita S3, consulta Gestione del ciclo di vita dello storage.

> **Note**
>
> Quando aggiungi una configurazione del ciclo di vita S3 a un bucket, Amazon S3 sostituisce l'attuale ciclo di vita del bucket, se presente. Per aggiornare una configurazione, la si recupera, si apportano le modifiche desiderate e successivamente si aggiunge la configurazione aggiornata al bucket.

L'esempio seguente mostra come utilizzare per aggiungere, aggiornare ed AWS SDK for Java eliminare la configurazione del ciclo di vita di un bucket. Inoltre, vengono effettuate le seguenti operazioni:

- Aggiunge una configurazione del ciclo di vita a un bucket
- Recupera la configurazione del ciclo di vita e la aggiorna aggiungendo un'altra regola.
- Aggiunge la configurazione del ciclo di vita modificata al bucket. Amazon S3 sostituisce la configurazione esistente.
- Recupera nuovamente la configurazione e verifica che abbia il numero corretto di regole stampando il numero di regole.
- Elimina la configurazione del ciclo di vita e verifica che sia stata eliminata cercando di recuperarla nuovamente.

Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
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

Puoi utilizzare il AWS SDK for .NET per gestire la configurazione del ciclo di vita di S3 su un bucket. Per ulteriori informazioni sulla gestione della configurazione del ciclo di vita, consulta Gestione del ciclo di vita dello storage.

> **Note**
>
> Quando aggiungi una configurazione del ciclo di vita, Amazon S3 sostituisce la configurazione esistente in un bucket specifico. Per aggiornare una configurazione del ciclo di vita esistente, è necessario prima di tutto recuperare la configurazione del ciclo di vita, apportare le modifiche e successivamente aggiungerla nel bucket.

L'esempio seguente mostra come utilizzare la configurazione del ciclo AWS SDK for .NET di vita di un bucket per aggiungere, aggiornare ed eliminare. L'esempio di codice esegue quanto segue:

- Aggiunge una configurazione del ciclo di vita a un bucket
- Recupera la configurazione del ciclo di vita e la aggiorna aggiungendo un'altra regola.
- Aggiunge la configurazione del ciclo di vita modificata al bucket. Amazon S3 sostituisce la configurazione del ciclo di vita esistente.
- Recupera nuovamente la configurazione e la verifica stampando il numero di regole nella configurazione.
- Elimina la configurazione del ciclo di vita e verifica l'eliminazione.

Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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

Puoi utilizzare AWS SDK for Ruby per gestire la configurazione del ciclo di vita di S3 su un bucket utilizzando la classe Configuration. AWS::S3::BucketLifecycle Per ulteriori informazioni sulla gestione della configurazione del ciclo di vita, consulta Gestione del ciclo di vita dello storage.

Utilizzo di REST API

Le sezioni seguenti della Documentazione d riferimento delle API di Amazon Simple Storage Service descrivono la REST API correlata alla configurazione del ciclo di vita S3.

- PutBucketLifecycleConfiguration
- GetBucketLifecycleConfiguration
- DeleteBucketLifecycle

Risoluzione dei problemi relativi al ciclo di vita di S3

Per i problemi comuni che potrebbero verificarsi quando si lavora con S3 Lifecycle, consulta. the section called “Risoluzione dei problemi del ciclo di vita”

Configurazioni del ciclo di vita e altre configurazioni del bucket

Oltre a quelle del ciclo di vita S3, è possibile associare altre configurazioni al bucket. In questa sezione viene descritto in che modo la configurazione del ciclo di vita S3 è correlata alle altre configurazioni del bucket.

Ciclo di vita e funzione Controllo delle versioni

Le configurazioni del ciclo di vita S3 possono essere aggiunte a bucket senza versione e a quelli che supportano la funzione Controllo delle versioni. Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Un bucket che supporta la funzione Controllo delle versioni mantiene una versione dell'oggetto corrente e zero o più versioni dell'oggetto non correnti. È possibile definire regole del ciclo di vita separate per le versioni dell'oggetto correnti e non correnti.

Per ulteriori informazioni, consulta Elementi della configurazione del ciclo di vita.

> **Important**
>
> Quando sono presenti più regole in una configurazione del ciclo di vita di S3, un oggetto può diventare idoneo per più azioni del ciclo di vita di S3 nello stesso giorno. In questi casi, Amazon S3 segue le seguenti regole generali:
>
> - L'eliminazione permanente ha la precedenza sul trasferimento.

- La transizione ha la precedenza sulla creazione di marker di eliminazione.
- Quando un oggetto è idoneo sia per una transizione S3 Glacier Flexible Retrieval che S3 Standard-IA (o S3 One Zone-IA), Amazon S3 sceglie la transizione S3 Glacier Flexible Retrieval.

Per alcuni esempi, consulta Esempio 5: Sovrapposizione di filtri, conflitto tra operazioni del ciclo di vita e comportamento di Amazon S3 con bucket senza versione.

Impostazione della configurazione del ciclo di vita in un bucket con MFA abilitata

La configurazione del ciclo di vita su bucket abilitati a più fattori (MFA) non è supportata.

Ciclo di vita e registrazione

Le azioni del ciclo di vita di Amazon S3 non vengono acquisite dalla AWS CloudTrail registrazione a livello di oggetto. CloudTrailacquisisce le richieste API effettuate agli endpoint Amazon S3 esterni, mentre le azioni del ciclo di vita di S3 vengono eseguite utilizzando endpoint Amazon S3 interni. I registri di accesso al server Amazon S3 possono essere abilitati in un bucket S3 per acquisire azioni relative al ciclo di vita S3, come la transizione di oggetti a un'altra classe di storage e la scadenza dell'oggetto con conseguente eliminazione permanente o eliminazione logica. Per ulteriori informazioni, consulta the section called “Registrazione dell'accesso al server”.

Se nel bucket è abilitata la registrazione, i log di accesso al server Amazon S3 segnalano i risultati delle seguenti operazioni:

Log dell'operazione	Descrizione
`S3.EXPIRE.OBJECT`	Amazon S3 elimina definitivamente l'oggetto a causa dell'azione di scadenza del ciclo di vita.
`S3.CREATE.DELETEMARKER`	Amazon S3 elimina logicamente la versione corrente e aggiunge un indicatore di eliminazi one in un bucket abilitato al controllo delle versioni.
`S3.TRANSITION_SIA.OBJECT`	Amazon S3 trasferisce l'oggetto nella classe di storage S3 Standard-IA.

Log dell'operazione	Descrizione
`S3.TRANSITION_ZIA.OBJECT`	Amazon S3 trasferisce l'oggetto nella classe di storage S3 One Zone – IA.
`S3.TRANSITION_INT.OBJECT`	Amazon S3 trasferisce l'oggetto nella classe di archiviazione S3 Intelligent-Tiering.
`S3.TRANSITION_GIR.OBJECT`	Amazon S3 avvia la transizione dell'oggetto alla classe di storage S3 Glacier Instant Retrieval.
`S3.TRANSITION.OBJECT`	Amazon S3 avvia la transizione dell'oggetto alla classe di storage S3 Glacier Flexible Retrieval.
`S3.TRANSITION_GDA.OBJECT`	Amazon S3 avvia la transizione dell'oggetto alla classe di storage S3 Glacier Deep Archive.
`S3.DELETE.UPLOAD`	Amazon S3 interrompe un caricamento incompleto in più parti.

> **Note**
>
> I registri dei log di accesso al server Amazon S3 sono generalmente inviati nel miglior modo possibile e non possono essere utilizzati per il resoconto completo di tutte le richieste di Amazon S3.

Risoluzione dei problemi relativi al ciclo di vita di S3

Per ulteriori informazioni sulla risoluzione dei problemi comuni relativi al ciclo di vita S3, consulta Risoluzione dei problemi del ciclo di vita di Amazon S3.

Ulteriori informazioni

- Elementi della configurazione del ciclo di vita
- Trasferimento nelle classi di archiviazione S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive (archiviazione di oggetti)
- Impostazione di una configurazione del ciclo di vita su un bucket

Configurazione delle notifiche di eventi del ciclo di vita

Puoi impostare una notifica di eventi Amazon S3 per ricevere un avviso quando Amazon S3 elimina un oggetto o lo trasferisce a un'altra classe di archiviazione Amazon S3 seguendo una regola del ciclo di vita S3.

Utilizzando i tipi di `LifecycleExpiration` eventi, puoi ricevere notifiche ogni volta che Amazon S3 elimina un oggetto in base alla tua configurazione del ciclo di vita di S3. Il tipo di evento `s3:LifecycleExpiration:Delete` avvisa quando viene eliminato un oggetto in un bucket senza versione. Notifica inoltre quando una versione dell'oggetto viene eliminata definitivamente da una configurazione del ciclo di vita S3. Il tipo di `s3:LifecycleExpiration:DeleteMarkerCreated` evento ti avvisa quando S3 Lifecycle crea un marker di eliminazione quando viene eliminata una versione corrente di un oggetto in un bucket con versioni. Per ulteriori informazioni, consulta Elimina versione dell'oggetto.

Utilizzando il tipo di `s3:LifecycleTransition` evento, puoi ricevere una notifica quando un oggetto viene trasferito da una classe di storage Amazon S3 a un'altra tramite una configurazione S3 Lifecycle.

Amazon S3 può pubblicare notifiche di eventi in un argomento Amazon Simple Notification Service (Amazon SNS), una coda Amazon Simple Queue Service (Amazon SQS) o una funzione AWS Lambda . Per ulteriori informazioni, consulta Notifiche di eventi Amazon S3.

Per istruzioni su come configurare Notifiche di eventi Amazon S3, consulta Abilitare le notifiche di eventi.

Quello che segue è un esempio di un messaggio inviato da Amazon S3 per pubblicare un evento `s3:LifecycleExpiration:Delete`. Per ulteriori informazioni, consulta Struttura del messaggio di 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":"example-s3-bucket",
               "ownerIdentity":{
                  "principalId":"A3NL1KOZZKExample"
               },
               "arn":"arn:aws:s3:::example-s3-bucket"
            },
            "object":{
               "key":"expiration/delete",
               "sequencer":"0055AED6DCD90281E5",
            }
         }
      }
   ]
}
```

I messaggi inviati da Amazon S3 per pubblicare un `s3:LifecycleTransition` evento includono anche le seguenti informazioni.

```
"lifecycleEventData":{
    "transitionEventData": {
        "destinationStorageClass": the destination storage class for the object
    }
}
```

Elementi della configurazione del ciclo di vita

Argomenti

- Elemento ID
- Elemento Status

- Elemento Filter
- Elementi per la descrizione delle operazioni nel ciclo di vita

È necessario specificare una configurazione del ciclo di vita di Amazon S3 in formato XML, costituita da una o più regole del ciclo di vita.

```
<LifecycleConfiguration>
    <Rule>
         ...
    </Rule>
    <Rule>
         ...
    </Rule>
</LifecycleConfiguration>
```

Ogni regola è costituita dagli elementi seguenti:

- Metadati delle regole che includono un ID della regola e uno stato che indica se la regola è abilitata o disabilitata. Se una regola è disabilitata, Amazon S3 non esegue le operazioni in essa specificate.
- Un filtro che identifica gli oggetti a cui si applica la regola. È possibile specificare un filtro utilizzando la dimensione dell'oggetto, il prefisso della chiave dell'oggetto, uno o più tag dell'oggetto o una combinazione di filtri.
- Una o più operazioni di transizione o scadenza con una data o un periodo nel ciclo di vita dell'oggetto in cui si desidera che Amazon S3 esegua l'operazione specificata.

Le sezioni che seguono descrivono gli elementi XML contenuti in una configurazione del ciclo di vita S3. Per gli esempi di configurazione, consulta Esempi di configurazione del ciclo di vita S3.

Elemento ID

Una configurazione del ciclo di vita S3 può contenere fino a 1.000 regole. Questo limite non è regolabile. L'`<ID>`elemento identifica in modo univoco una regola. La lunghezza dell'ID è limitata a 255 caratteri.

Elemento Status

Il valore <Status> dell'elemento può essere oEnabled. Disabled Se una regola è disabilitata, Amazon S3 non esegue nessuna delle operazioni in essa definite.

Elemento Filter

Una regola del ciclo di vita può essere applicata a tutti o a un sottoinsieme di oggetti in un bucket in base all'<Filter>elemento specificato nella regola del ciclo di vita.

È possibile filtrare gli oggetti per prefisso della chiave, tag dell'oggetto o una combinazione di entrambi (in tal caso, Amazon S3 utilizza un operatore logico AND per combinare i filtri). Considerare i seguenti esempi:

- Specificazione di un filtro utilizzando i prefissi chiave: questo esempio mostra una regola del ciclo di vita S3 che si applica a un sottoinsieme di oggetti in base al prefisso del nome chiave (). logs/ Ad esempio, la regola del ciclo di vita si applica agli oggetti, e. logs/mylog.txt logs/temp1.txt logs/test.txt ma non per l'oggetto example.jpg.

  ```
  <LifecycleConfiguration>
      <Rule>
          <Filter>
             <Prefix>logs/</Prefix>
          </Filter>
          transition/expiration actions
           ...
      </Rule>
      ...
  </LifecycleConfiguration>
  ```

  Se desiderate applicare un'azione del ciclo di vita a un sottoinsieme di oggetti in base a prefissi di nomi chiave diversi, specificate regole separate. all'interno di ognuna, specificare un filtro basato sul prefisso. Ad esempio, per descrivere un'azione del ciclo di vita per gli oggetti con i prefissi chiave e, specificate projectA/ due regole come segue: projectB/

  ```
  <LifecycleConfiguration>
      <Rule>
          <Filter>
             <Prefix>projectA/</Prefix>
          </Filter>
          transition/expiration actions
  ```

```
        ...
    </Rule>

    <Rule>
        <Filter>
           <Prefix>projectB/</Prefix>
        </Filter>
        transition/expiration actions
        ...
    </Rule>
</LifecycleConfiguration>
```

Per ulteriori informazioni sulle chiavi degli oggetti, consulta Creazione di nomi di chiavi oggetto.

- Specificazione di un filtro basato sui tag degli oggetti: nell'esempio seguente, la regola del ciclo di vita specifica un filtro basato su un tag () e un valore (). *key value* La regola si applica quindi solo a un sottoinsieme di oggetti con quel tag specifico.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Tag>
              <Key>key</Key>
              <Value>value</Value>
           </Tag>
        </Filter>
        transition/expiration actions
        ...
    </Rule>
</LifecycleConfiguration>
```

È possibile specificare un filtro basato su più tag. È necessario racchiudere i tag nell'<And>elemento, come illustrato nell'esempio seguente. La regola indica ad Amazon S3 di eseguire le operazioni del ciclo di vita sugli oggetti con due tag (con la specifica combinazione di chiave e valore).

```
<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 regola del ciclo di vita si applica agli oggetti che contengono entrambi i tag specificati. Amazon S3 esegue un'operazione logica AND. Tieni presente quanto segue:

- Ogni tag deve corrispondere esattamente alla chiave e al valore. Se specificate solo un <Key> elemento e nessun <Value> elemento, la regola si applicherà solo agli oggetti che corrispondono alla chiave del tag e che non hanno un valore specificato.
- La regola si applica a un sottoinsieme di oggetti che ha tutti i tag specificati nella regola. Se a un oggetto sono specificati tag aggiuntivi, la regola verrà comunque applicata.

> **Note**
>
> Quando si specificano più tag in un filtro, ogni chiave del tag deve essere univoca.

- Specificazione di un filtro basato sia sul prefisso che su uno o più tag: in una regola del ciclo di vita, puoi specificare un filtro basato sia sul prefisso chiave che su uno o più tag. Anche in questo caso, è necessario racchiudere tutti questi elementi del filtro nell'elemento, come segue<And>:

```
<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 questi filtri utilizzando una logica. AND Cioè, la regola si applica al sottoinsieme di oggetti con il prefisso chiave specificato e i tag specificati. Un filtro può avere un solo prefisso e zero o più tag.

- È possibile specificare un filtro vuoto, nel qual caso la regola si applica a tutti gli oggetti nel bucket.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

- Per filtrare una regola per dimensione oggetto, è possibile specificare una dimensione minima (`ObjectSizeGreaterThan`) o massima (`ObjectSizeLessThan`) oppure è possibile specificare un intervallo di dimensioni degli oggetti.

  I valori delle dimensioni degli oggetti sono espressi in byte. La dimensione massima del filtro è di 5 TB. Alcune classi di archiviazione hanno limiti minimi di dimensione degli oggetti. Per ulteriori informazioni, consulta Confronto delle classi di storage di Amazon S3.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
            <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

> **Note**
>
> I `ObjectSizeLessThan` filtri `ObjectSizeGreaterThan` and escludono i valori specificati. Ad esempio, se imposti oggetti con dimensioni da 128 KB a 1024 KB per spostarli dalla classe di storage S3 Standard alla classe di storage S3 Standard-IA, gli oggetti che sono esattamente 1024 KB e 128 KB non passeranno a S3 Standard-IA. La regola si applicherà invece solo agli oggetti con dimensioni superiori a 128 KB e inferiori a 1024 KB.

Se stai specificando un intervallo di dimensioni degli oggetti, il numero intero `ObjectSizeGreaterThan` deve essere minore del valore `ObjectSizeLessThan`. Quando si utilizzano più filtri, è necessario racchiudere i filtri in un elemento <And>. L'esempio seguente mostra come specificare oggetti in un intervallo compreso tra 500 byte e 64.000 byte.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
            <And>
                <Prefix>key-prefix</Prefix>
                <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
                <ObjectSizeLessThan>64000</ObjectSizeLessThan>
            </And>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

Elementi per la descrizione delle operazioni nel ciclo di vita

È possibile indicare ad Amazon S3 di eseguire operazioni specifiche nel ciclo di vita di un oggetto specificando in una regola del ciclo di vita S3 una o più delle seguenti operazioni predefinite, il cui effetto dipende dallo stato della funzione Controllo delle versioni del bucket.

- **`Transition`**elemento di azione: si specifica l'`Transition`azione per la transizione degli oggetti da una classe di archiviazione all'altra. Per ulteriori informazioni sul trasferimento degli oggetti,

consulta Trasferimenti supportati e vincoli correlati. Al raggiungimento di una data o di un periodo nel ciclo di vita dell'oggetto, Amazon S3 esegue la transizione.

Per un bucket con versione (bucket con funzione Controllo delle versioni attivata o sospesa), l'operazione `Transition` si applica alla versione corrente dell'oggetto. Per la gestione delle versioni non correnti, Amazon S3 definisce l'operazione `NoncurrentVersionTransition` (descritta di seguito in questo argomento).

- **`Expiration`**elemento di azione: l'`Expiration`azione fa scadere gli oggetti identificati nella regola e si applica agli oggetti idonei in qualsiasi classe di storage Amazon S3. Per ulteriori informazioni sulle classi di storage, consulta Utilizzo delle classi di storage di Amazon S3. Amazon S3 rende non disponibili tutti gli oggetti scaduti. L'eventuale rimozione permanente degli oggetti dipende dallo stato della funzione Controllo delle versioni del bucket.
  - Bucket senza versione: l'`Expiration`azione comporta la rimozione permanente dell'oggetto da parte di Amazon S3.
  - Bucket con versione – Per un bucket con versione (ovvero, con funzione Controllo delle versioni attivata o sospesa), sono diversi i fattori che governano la gestione dell'operazione `Expiration` da parte di Amazon S3. Per i bucket con controllo delle versioni abilitata o sospesa, vale quanto segue:
    - L'operazione `Expiration` si applica solo alla versione corrente (non ha effetto sulle versioni non correnti dell'oggetto).
    - Amazon S3 non esegue alcuna operazione se sono presenti una o più versioni dell'oggetto e il contrassegno di eliminazione è la versione corrente.
    - Se la versione corrente dell'oggetto è l'unica disponibile e porta anche il contrassegno di eliminazione (noto anche come contrassegno di eliminazione dell'oggetto scaduto, dove tutte le versioni degli oggetti vengono eliminate e rimane solo un contrassegno di eliminazione), Amazon S3 rimuove il contrassegno di eliminazione dall'oggetto scaduto. È possibile inoltre utilizzare l'operazione di eliminazione per indicare ad Amazon S3 di rimuovere i contrassegni di eliminazione dell'oggetto scaduto. Per un esempio, consulta Esempio 7: rimozione dei contrassegni di eliminazione oggetto scaduto.

    Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

    Considera inoltre quanto segue durante la configurazione di Amazon S3 per la gestione delle scadenze:

    - Bucket con funzione Controllo delle versioni abilitata

Se la versione dell'oggetto corrente non è un contrassegno di eliminazione, Amazon S3 aggiunge un contrassegno di eliminazione con un ID versione univoco. La versione corrente diventa quindi non corrente e il contrassegno di eliminazione diventa la versione corrente.

- Bucket con funzione Controllo delle versioni sospesa

  In un bucket con versione sospesa, l'azione di scadenza fa sì che Amazon S3 crei un marker di eliminazione con come ID di versione. `null` Il contrassegno di eliminazione sostituisce qualsiasi versione dell'oggetto con ID versione null nella gerarchia delle versioni: questa operazione elimina di fatto l'oggetto.

Inoltre, in Amazon S3 sono disponibili le seguenti operazioni, utili per gestire le versioni non correnti degli oggetti in un bucket con versione (ovvero, con funzione Controllo delle versioni attivata o sospesa).

- **`NoncurrentVersionTransition`**elemento di azione: utilizza questa azione per specificare quando Amazon S3 esegue la transizione degli oggetti alla classe di storage specificata. Puoi basare questa scadenza su un certo numero di giorni da quando gli oggetti sono diventati non correnti. Oltre al numero di giorni, puoi anche fornire un numero massimo di versioni non correnti da conservare (compreso tra 1 e 100). Questo valore determina quante versioni non correnti più recenti devono esistere prima che Amazon S3 possa eseguire l'azione associata su una determinata versione. Amazon S3 trasferirà eventuali versioni aggiuntive non correnti oltre il numero specificato da conservare.

  Per specificare il numero massimo di versioni non correnti, devi fornire anche un elemento. `Filter` Se non specifichi un `Filter` elemento, Amazon S3 genera un `InvalidRequest` errore quando fornisci un numero massimo di versioni non correnti.

  Per ulteriori informazioni sul trasferimento degli oggetti, consulta Trasferimenti supportati e vincoli correlati. Per informazioni dettagliate su come Amazon S3 calcola la data quando si specifica il numero di giorni nell'operazione `NoncurrentVersionTransition`, consulta Regole del ciclo di vita basate sull'età di un oggetto.
- **`NoncurrentVersionExpiration`**elemento di azione: utilizza questa azione per fare in modo che Amazon S3 elimini definitivamente le versioni non correnti degli oggetti. Questi oggetti eliminati non possono essere recuperati. È possibile basare questa scadenza su un determinato numero di giorni trascorsi da quando gli oggetti non sono più correnti. Oltre al numero di giorni, puoi anche fornire un numero massimo di versioni non correnti da conservare (compreso tra 1 e 100).

Questo valore specifica quante versioni non correnti più recenti devono esistere prima che Amazon S3 possa eseguire l'operazione associata su una determinata versione. Amazon S3 eliminerà definitivamente tutte le versioni non correnti aggiuntive oltre al numero specificato da conservare.

Per specificare il numero massimo di versioni non correnti, devi fornire anche un elemento. `Filter` Se non specifichi un `Filter` elemento, Amazon S3 genera un `InvalidRequest` errore quando fornisci un numero massimo di versioni non correnti.

Il ritardo nella rimozione degli oggetti non correnti può rivelarsi utile per correggere eventuali eliminazioni o sovrascritture accidentali. Ad esempio, è possibile configurare una regola di scadenza perché elimini le versioni cinque giorni dopo essere diventate non correnti. Ad esempio, supponiamo che il 01/01/2014 alle 10:30 UTC, crei un oggetto chiamato (ID versione 111111). `photo.gif` Il 02/01/2014 alle 11:30 UTC, si elimina accidentalmente (ID versione 111111), il che crea un indicatore di eliminazione con un nuovo ID di versione `photo.gif` (ad esempio l'ID di versione 4857693). A questo punto, si hanno cinque giorni per recuperare la versione originale di `photo.gif` (ID versione 111111) prima che l'eliminazione diventi definitiva. Il 1° agosto 2014 alle 00:00 UTC, la regola di scadenza del ciclo di vita viene eseguita ed `photo.gif` eliminata definitivamente (ID versione 111111), cinque giorni dopo che è diventata una versione non corrente.

Per informazioni dettagliate su come Amazon S3 calcola la data quando si specifica il numero di giorni in un'operazione `NoncurrentVersionExpiration`, consulta Regole del ciclo di vita basate sull'età di un oggetto.

> **Note**
>
> Le configurazioni del ciclo di vita con scadenza degli oggetti non rimuovono i caricamenti incompleti in più parti. Per rimuovere i caricamenti multiparte incompleti, è necessario utilizzare l'azione di configurazione del `AbortIncompleteMultipartUpload` ciclo di vita descritta più avanti in questa sezione.

Oltre alle azioni di transizione e scadenza, puoi utilizzare le seguenti azioni di configurazione del ciclo di vita per fare in modo che Amazon S3 interrompa i caricamenti multiparte incompleti o rimuova i marker di eliminazione degli oggetti scaduti:

- **`AbortIncompleteMultipartUpload`**elemento di azione: utilizza questo elemento per impostare il tempo massimo (in giorni) durante il quale desideri che i caricamenti in più parti

rimangano in corso. Se i caricamenti multiparte applicabili (determinati dal nome chiave `prefix` specificato nella regola del ciclo di vita) non vengono completati correttamente entro il periodo di tempo predefinito, Amazon S3 interrompe i caricamenti multiparte incompleti. Per ulteriori informazioni, consulta Interruzione di un caricamento in più parti.

> **Note**
>
> Non puoi specificare questa azione del ciclo di vita in una regola con un filtro che utilizza tag di oggetto.

- **`ExpiredObjectDeleteMarker`**elemento di azione: in un bucket abilitato al controllo delle versioni, un marker di eliminazione con zero versioni non correnti viene definito marker di eliminazione di oggetti scaduti. Puoi utilizzare questa azione del ciclo di vita per indirizzare Amazon S3 a rimuovere i marker di eliminazione degli oggetti scaduti. Per vedere un esempio, consulta Esempio 7: rimozione dei contrassegni di eliminazione oggetto scaduto.

> **Note**
>
> Non puoi specificare questa azione del ciclo di vita in una regola con un filtro che utilizza i tag degli oggetti.

Modalità con cui Amazon S3 calcola da quanto tempo un oggetto è non corrente

In un bucket abilitato al controllo delle versioni, possono essere incluse più versioni di un oggetto. Esiste sempre una versione corrente e zero o più versioni non correnti. Ogni volta che si carica un oggetto, la versione appena aggiunta diventa la versione corrente e quella che lo era in precedenza viene mantenuta come versione non corrente. Per stabilire da quanti giorni è non corrente un oggetto, Amazon S3 considera la data di creazione della versione successiva. Il numero di giorni dalla data di creazione della versione successiva viene utilizzato da Amazon S3 come numero di giorni da cui l'oggetto è non corrente.

> **Ripristino delle versioni precedenti di un oggetto con l'uso delle configurazioni del ciclo di vita di S3**
>
> Come spiegato inRipristino di versioni precedenti, è possibile utilizzare uno dei due metodi seguenti per recuperare le versioni precedenti di un oggetto:

- Metodo 1 — Copia una versione non corrente dell'oggetto nello stesso bucket. La copia diventa la versione corrente dell'oggetto e vengono conservate tutte le sue versioni.
- Metodo 2 — Eliminare definitivamente la versione corrente dell'oggetto. Così facendo, in effetti, la versione prima non corrente diventa la versione corrente dell'oggetto.

Quando utilizzi le regole di configurazione di S3 Lifecycle con bucket abilitati al controllo delle versioni, come best practice consigliamo di utilizzare il Metodo 1.
Il ciclo di vita di S3 opera in base a un modello consistente finale. Una versione corrente che hai eliminato definitivamente potrebbe non scomparire finché le modifiche non si propagano a tutti i sistemi Amazon S3. (Pertanto, Amazon S3 potrebbe non essere temporaneamente a conoscenza di questa eliminazione.) Nel frattempo. la regola del ciclo di vita configurata per la scadenza degli oggetti non correnti potrebbe rimuovere definitivamente tali oggetti, compreso quello da ripristinare. Quindi, copiare la vecchia versione, come consigliato nel Metodo 1, è l'alternativa più sicura.

Operazioni del ciclo di vita e stato della funzione Controllo delle versioni nel bucket

Regole del ciclo di vita basate sull'età di un oggetto

Puoi specificare un periodo di tempo, nel numero di giorni dalla creazione (o modifica) dell'oggetto, in cui Amazon S3 può intraprendere l'azione specificata.

Quando si specifica il numero di giorni nelle operazioni `Transition` ed `Expiration` in una configurazione del ciclo di vita S3, tenere presente quanto segue:

- Il valore specificato è il numero di giorni trascorsi dalla creazione dell'oggetto in cui verrà eseguita l'azione.
- Amazon S3 calcola il tempo aggiungendo il numero di giorni specificato nella regola all'ora di creazione dell'oggetto e arrotondando l'ora risultante al giorno successivo a mezzanotte UTC. Ad esempio, se un oggetto è stato creato il 15/01/2014 alle 10:30 UTC e specifichi 3 giorni in una regola di transizione, la data di transizione dell'oggetto verrà calcolata come 19/01/2014 00:00 UTC.

> **Note**
>
> Amazon S3 gestisce solo l'ultima data di modifica per ciascun oggetto. Ad esempio, la console Amazon S3 mostra la data dell'ultima modifica nel riquadro Proprietà dell'oggetto. Quando crei inizialmente un nuovo oggetto, questa data riflette la data di creazione dell'oggetto. Se l'oggetto viene sostituito, la data cambia di conseguenza. Pertanto, la data di creazione è sinonimo della data dell'ultima modifica.

Quando si specifica il numero di giorni nelle operazioni `NoncurrentVersionTransition` e `NoncurrentVersionExpiration` in una configurazione del ciclo di vita, tenere presente quanto segue:

- Il valore specificato è il numero di giorni dal momento in cui la versione dell'oggetto diventa non corrente (ovvero, quando l'oggetto viene sovrascritto o eliminato) in cui Amazon S3 eseguirà l'azione sull'oggetto o sugli oggetti specificati.
- Amazon S3 calcola l'ora aggiungendo il numero di giorni specificato nella regola all'ora in cui viene creata la nuova versione successiva dell'oggetto e arrotondando l'ora risultante al giorno successivo a mezzanotte UTC. Ad esempio, nel tuo bucket, supponiamo di avere una versione corrente di un oggetto creato il 01/01/2014 alle 10:30 UTC. Se la nuova versione dell'oggetto che sostituisce la versione corrente viene creata il 15/01/2014 alle 10:30 UTC e specificate 3 giorni in una regola di transizione, la data di transizione dell'oggetto viene calcolata come 19/01/2014 00:00 UTC.

Regole del ciclo di vita basate su una data specifica

Quando si specifica un'operazione in una regola del ciclo di vita S3, è possibile specificare la data in cui si desidera che Amazon S3 esegua l'operazione. Alla data specificata, Amazon S3 applica l'operazione a tutti gli oggetti idonei (in base ai criteri di filtro).

Se si specifica un'azione del ciclo di vita S3 con una data precedente, tutti gli oggetti qualificati diventano immediatamente idonei per tale azione del ciclo di vita.

> **Important**
>
> Le operazioni basate su data non sono valide una tantum. Se lo stato della regola è , Amazon S3 continua ad applicare l'operazione basata su data anche dopo che questa è trascors `Enabled`.

Ad esempio, supponiamo di specificare un'`Expiration`azione basata sulla data per eliminare tutti gli oggetti (supponiamo che nella regola non sia specificato alcun filtro). Nella data specificata, Amazon S3 applica la scadenza a tutti gli oggetti nel bucket. Amazon S3 continua inoltre a far scadere tutti i nuovi oggetti creati nel bucket. Per interrompere l'azione del ciclo di vita, devi rimuovere l'azione dalla regola del ciclo di vita, disabilitare la regola o eliminare la regola dalla configurazione del ciclo di vita.

Il valore della data deve essere conforme allo standard ISO 8601. L'ora è sempre la mezzanotte UTC.

> **Note**
>
> Non puoi creare regole del ciclo di vita basate sulla data utilizzando la console Amazon S3, ma puoi visualizzare, disabilitare o eliminare tali regole.

Esempi di configurazione del ciclo di vita S3

In questa sezione vengono forniti alcuni esempi di configurazione del ciclo di vita S3. Ogni esempio mostra come si può specificare il codice XML in ciascun scenario di esempio.

Argomenti

Esempio 1: specifica di un filtro

Ogni regola del ciclo di vita S3 include un filtro che è possibile utilizzare per identificare un sottoinsieme di oggetti nel bucket a cui si applica la regola del ciclo di vita S3. Le configurazioni del ciclo di vita S3 seguenti mostrano esempi di come specificare un filtro.

- In questa regola di configurazione del ciclo di vita S3, il filtro specifica un prefisso della chiave (`tax/`). Pertanto la regola si applica agli oggetti con il prefisso del nome della chiave `tax/`, ad esempio `tax/doc1.txt` e `tax/doc2.txt`.

  La regola specifica due operazioni che richiedono ad Amazon S3 di eseguire quanto segue:

  - Trasferire gli oggetti nella classe di archiviazione S3 Glacier Flexible Retrieval dopo 365 giorni (un anno) dalla data di creazione.
  - Eliminare gli oggetti (operazione `Expiration`) dopo 3.650 giorni (10 anni) dalla data di creazione.

  ```
  <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>
  ```

  Invece di specificare l'età dell'oggetto in termini di giorni dopo la creazione, puoi specificare una data per ogni azione. Non è tuttavia possibile utilizzare sia `Date` sia `Days` nella stessa regola.

- Per applicare la regola del ciclo di vita S3 a tutti gli oggetti nel bucket, occorre specificare un prefisso vuoto. Nella configurazione seguente, la regola specifica un'operazione `Transition` che indica ad Amazon S3 di trasferire gli oggetti nella classe di archiviazione S3 Glacier Flexible Retrieval dopo 0 giorni dalla data di creazione. Questa regola significa che gli oggetti sono idonei

per l'archiviazione su S3 Glacier Flexible Retrieval a mezzanotte UTC dopo la creazione. Per ulteriori informazioni sui vincoli del ciclo di vita, consulta la sezione Vincoli.

```
<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>
```

- È possibile specificare zero o un prefisso del nome della chiave e zero o più tag di oggetto in un filtro. Nel codice di esempio riportato di seguito la regola del ciclo di vita S3 viene applicata a un sottoinsieme di oggetti con il prefisso della chiave `tax/` e agli oggetti che dispongono di due tag con chiave e valore specifici. Quando si specifica più di un filtro, è necessario includere l'elemento <And> come mostrato (Amazon S3 applica un'istruzione AND logica per combinare le condizioni del filtro specificate).

```
...
<Filter>
   <And>
      <Prefix>tax/</Prefix>
      <Tag>
         <Key>key1</Key>
         <Value>value1</Value>
      </Tag>
      <Tag>
         <Key>key2</Key>
         <Value>value2</Value>
      </Tag>
    </And>
</Filter>
...
```

- È possibile filtrare gli oggetti in base solo ai tag. La regola dei ciclo di vita S3 riportata di seguito, ad esempio, viene applicata agli oggetti che dispongono dei due tag specificati (non viene specificato alcun prefisso).

```
...
<Filter>
   <And>
      <Tag>
         <Key>key1</Key>
         <Value>value1</Value>
      </Tag>
      <Tag>
         <Key>key2</Key>
         <Value>value2</Value>
      </Tag>
    </And>
</Filter>
...
```

Important

Quando sono presenti più regole in una configurazione S3 Lifecycle, un oggetto può diventare idoneo per più azioni S3 Lifecycle nello stesso giorno. In questi casi, Amazon S3 segue le seguenti regole generali:

- L'eliminazione permanente ha la precedenza sul trasferimento.
- La transizione ha la precedenza sulla creazione di marker di eliminazione.
- Quando un oggetto è idoneo sia per una transizione S3 Glacier Flexible Retrieval che per S3 Standard-IA (o S3 One Zone-IA), Amazon S3 sceglie la transizione S3 Glacier Flexible Retrieval.

Per alcuni esempi, consulta Esempio 5: Sovrapposizione di filtri, conflitto tra operazioni del ciclo di vita e comportamento di Amazon S3 con bucket senza versione.

Esempio 2: Disabilitare una regola del ciclo di vita

Puoi disabilitare temporaneamente una regola del ciclo di vita S3. Nella configurazione del ciclo di vita S3 seguente sono specificate due regole:

- La regola 1 indica ad Amazon S3 di trasferire gli oggetti con il prefisso `logs/` nella classe di archiviazione S3 Glacier Flexible Retrieval subito dopo la creazione.
- La regola 2 indica ad Amazon S3 di trasferire gli oggetti con il prefisso `documents/` nella classe di archiviazione S3 Glacier Flexible Retrieval subito dopo la creazione.

Nella configurazione, la regola 1 è abilitata e la regola 2 è disabilitata. Amazon S3 ignora le regole disabilitate.

```
<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>
```

Esempio 3: suddivisione della classe di storage su più livelli per tutta la durata di un oggetto

In questo esempio, la configurazione del ciclo di vita S3 viene utilizzata per abbassare la classe di archiviazione degli oggetti nel corso della loro esistenza. Tale abbassamento contribuisce a ridurre i costi di storage. Per ulteriori informazioni sui prezzi, consulta la sezione Prezzi di Amazon S3.

Nella configurazione del ciclo di vita S3 riportata di seguito viene specificata una regola che si applica agli oggetti con il prefisso del nome della chiave `logs/`. Nella regola vengono specificate le seguenti operazioni:

- Due operazioni di trasferimento:
  - Trasferimento degli oggetti nella classe di archiviazione S3 Standard-IA - accesso infrequente dopo 30 giorni dalla data di creazione.
  - Trasferimento degli oggetti nella classe di archiviazione S3 Glacier Flexible Retrieval dopo 90 giorni dalla data di creazione.
- Una operazione di scadenza che indica ad Amazon S3 di eliminare questi oggetti un anno dopo la creazione.

```
<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**
>
> È possibile utilizzare un'unica regola per descrivere tutte le operazioni del ciclo di vita S3, se queste si applicano allo stesso insieme di oggetti (identificati dal filtro). In caso contrario, si possono aggiungere più regole per ogni oggetto specificando un filtro diverso.

> **Important**
>
> Quando sono presenti più regole in una configurazione del ciclo di vita di S3, un oggetto può diventare idoneo per più azioni del ciclo di vita di S3 nello stesso giorno. In questi casi, Amazon S3 segue le seguenti regole generali:
>
> - L'eliminazione permanente ha la precedenza sul trasferimento.
> - La transizione ha la precedenza sulla creazione di marker di eliminazione.
> - Quando un oggetto è idoneo sia per una transizione S3 Glacier Flexible Retrieval che per S3 Standard-IA (o S3 One Zone-IA), Amazon S3 sceglie la transizione S3 Glacier Flexible Retrieval.
>
> Per alcuni esempi, consulta Esempio 5: Sovrapposizione di filtri, conflitto tra operazioni del ciclo di vita e comportamento di Amazon S3 con bucket senza versione.

Esempio 4: specifica di più regole

Per eseguire operazioni del ciclo di vita S3 diverse su vari oggetti, è possibile specificare più regole. Nella configurazione del ciclo di vita S3 seguente sono specificate due regole:

- La regola 1 si applica agli oggetti con il prefisso nel nome della chiav `classA/`. Questa regola indica ad Amazon S3 di trasferire gli oggetti alla classe di archiviazione S3 Glacier Flexible Retrieval un anno dopo la creazione e di rimuoverli dopo 10 anni dalla creazione.
- La regola 2 si applica agli oggetti con il prefisso nel nome della chiav `classB/`. Questa regola indica ad Amazon S3 di trasferire gli oggetti alla classe di storage S3 Standard-IA 90 giorni dopo la creazione e di eliminarli dopo un anno dalla creazione.

```
<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

Quando sono presenti più regole in una configurazione S3 Lifecycle, un oggetto può diventare idoneo per più azioni S3 Lifecycle nello stesso giorno. In questi casi, Amazon S3 segue le seguenti regole generali:

- L'eliminazione permanente ha la precedenza sul trasferimento.
- La transizione ha la precedenza sulla creazione di marker di eliminazione.
- Quando un oggetto è idoneo sia per una transizione S3 Glacier Flexible Retrieval che per S3 Standard-IA (o S3 One Zone-IA), Amazon S3 sceglie la transizione S3 Glacier Flexible Retrieval.

Per alcuni esempi, consulta Esempio 5: Sovrapposizione di filtri, conflitto tra operazioni del ciclo di vita e comportamento di Amazon S3 con bucket senza versione.

Esempio 5: Sovrapposizione di filtri, conflitto tra operazioni del ciclo di vita e comportamento di Amazon S3 con bucket senza versione

In una configurazione del ciclo di vita S3 potrebbero venire specificati prefissi o operazioni che si sovrappongono.

In genere, il ciclo di vita S3 ottimizza i costi. Ad esempio, se due policy per la scadenza di sovrappongono, verrà onorata la policy per la scadenza più breve in modo che i dati non rimangano archiviati più a lungo del previsto. Analogamente, se due policy di trasferimento si sovrappongono, il ciclo di vita S3 eseguirà la transizione degli oggetti nella classe di storage ottimizzata per i costi.

In entrambi i casi, il ciclo di vita S3 tenta di scegliere il percorso meno costoso per l'utente. Un'eccezione a questa regola generale è con la classe di storage S3 Intelligent-Tiering. S3 Intelligent-Tiering è preferito dal ciclo di vita S3 rispetto a qualsiasi classe di archiviazione, a parte S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive.

Negli esempi seguenti viene illustrato in che modo Amazon S3 sceglie di risolvere i potenziali conflitti.

Example 1: sovrapposizione di prefissi (nessun conflitto)

La configurazione di esempio riportata di seguito include due regole in cui sono specificati prefissi che si sovrappongono nel modo seguente:

- La prima regola specifica un filtro vuoto, che indica tutti gli oggetti nel bucket.
- La seconda regola specifica un prefisso del nome della chiave (`logs/`), che indica solo un sottoinsieme di oggetti.

La regola 1 richiede ad Amazon S3 di eliminare tutti gli oggetti un anno dopo la creazione. La regola 2 richiede ad Amazon S3 di passare un sottoinsieme di oggetti alla classe di storage S3 Standard-IA 30 giorni dopo la creazione.

```
<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>
```

Poiché in questo caso non vi sono conflitti, Amazon S3 trasferirà gli oggetti con il prefisso `logs/` nella classe di archiviazione S3 Standard-IA dopo 30 giorni dalla data di creazione. Una volta passato un anno dalla data di creazione di qualsiasi oggetto, questo verrà eliminato.

Example 2: conflitto tra operazioni del ciclo di vita

Questa configurazione di esempio include due regole che indicano ad Amazon S3 di eseguire due operazioni diverse sullo stesso insieme di oggetti nello stesso momento dell'esistenza degli oggetti:

- Entrambe le regole specificano lo stesso prefisso nel nome della chiave, quindi entrambe le regole si applicano allo stesso insieme di oggetti.
- Entrambe le regole specificano che devono essere applicate 365 giorni dopo la data di creazione degli oggetti.
- Una regola indica ad Amazon S3 di eseguire la transizione degli oggetti alla classe di storage S3 Standard-IA, mentre un'altra specifica che Amazon S3 deve fare scadere gli oggetti contemporaneamente.

```
<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>
```

In questo caso, dal momento che l'obiettivo è di far scadere (rimuovere) gli oggetti, non ha senso cambiare la classe di archiviazione, quindi Amazon S3 sceglie semplicemente di eseguire l'operazione di scadenza su questi oggetti.

Example 3: sovrapposizione di prefissi con conseguente conflitto tra operazioni del ciclo di vita

In questo esempio la configurazione include due regole, in cui sono specificati prefissi che si sovrappongono nel modo seguente:

- La regola 1 specifica un prefisso vuoto, che indica tutti gli oggetti.
- La regola 2 specifica un prefisso della chiave (`logs/`), che indica un sottoinsieme di tutti gli oggetti.

Per il sottoinsieme di oggetti con il prefisso nel nome della chiave `logs/`, si applicano le operazioni del ciclo di vita S3 in entrambe le regole. Una regola indica ad Amazon S3 di trasferire gli oggetti 10 giorni dopo la data di creazione mentre un'altra regola indica ad Amazon S3 di trasferirli 365 giorni dopo la data di creazione.

```
<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>
```

In questo caso, Amazon S3 sceglie di trasferirli 10 giorni dopo la data di creazione.

Example 4: applicazione di un filtro basato su tag con conseguente conflitto tra operazioni del ciclo di vita

Supponiamo di avere la configurazione del ciclo di vita S3 seguente con due regole, in ognuna delle quali è specificato un filtro basato su tag:

- La regola 1 specifica un filtro basato su tag (`tag1/value1`). Questa regola indica ad Amazon S3 di trasferire gli oggetti nella classe di archiviazione S3 Glacier Flexible Retrieval 365 giorni dopo la data di creazione.
- La regola 2 specifica un filtro basato su tag (`tag2/value2`). Questa regola indica ad Amazon S3 di far scadere gli oggetti 14 giorni dopo la creazione.

Nell'esempio di seguito viene mostrata la configurazione del ciclo di vita S3.

```
<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>
```

Se un oggetto ha entrambi i tag, Amazon S3 deve decidere quale regola seguire. In questo caso, Amazon S3 fa scadere l'oggetto 14 giorni dopo la data di creazione. L'oggetto viene rimosso, quindi non viene eseguita l'operazione di trasferimento.

Important

Quando sono presenti più regole in una configurazione S3 Lifecycle, un oggetto può diventare idoneo per più azioni S3 Lifecycle nello stesso giorno. In questi casi, Amazon S3 segue le seguenti regole generali:

- L'eliminazione permanente ha la precedenza sul trasferimento.
- La transizione ha la precedenza sulla creazione di marker di eliminazione.

- Quando un oggetto è idoneo sia per una transizione S3 Glacier Flexible Retrieval che per S3 Standard-IA (o S3 One Zone-IA), Amazon S3 sceglie la transizione S3 Glacier Flexible Retrieval.

Per alcuni esempi, consulta Esempio 5: Sovrapposizione di filtri, conflitto tra operazioni del ciclo di vita e comportamento di Amazon S3 con bucket senza versione.

Esempio 6: specifica di una regola del ciclo di vita per un bucket che supporta la funzione Controllo delle versioni

Supponiamo di avere un bucket con il controllo delle versioni abilitato. Questo significa che per ogni oggetto esistono una versione corrente e zero o più versioni non correnti. (Per ulteriori informazioni su Controllo versioni S3, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.)
In questo esempio si vuole mantenere la cronologia di un anno e quindi eliminare le versioni non correnti. Le configurazioni S3 Lifecycle supportano il mantenimento da 1 a 100 versioni di qualsiasi oggetto.

Per risparmiare sui costi di archiviazione è necessario spostare le versioni non correnti in S3 Glacier Flexible Retrieval 30 giorni dopo che diventano non correnti (supponendo che si tratti di dati cold a cui non è necessario accedere in tempo reale). Inoltre, ti aspetti che la frequenza di accesso delle versioni correnti diminuisca 90 giorni dopo la creazione, quindi potresti scegliere di spostare questi oggetti nella classe di storage 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>
```

Esempio 7: rimozione dei contrassegni di eliminazione oggetto scaduto

Un bucket abilitato per la funzione Controllo delle versioni mantiene una versione corrente e zero o più versioni non correnti di ogni oggetto. Quando si elimina un oggetto, tenere presente quanto segue:

- Se non si specifica un ID versione nella richiesta di eliminazione, Amazon S3 aggiunge un contrassegno di eliminazione invece di eliminare l'oggetto. La versione dell'oggetto corrente diventa non corrente, quindi il contrassegno di eliminazione diventa la versione corrente.
- Se si specifica un ID versione nella richiesta di eliminazione, Amazon S3 elimina la versione dell'oggetto in modo permanente (non viene creato alcun contrassegno di eliminazione).
- Un contrassegno di eliminazione con zero versioni non correnti viene definito un contrassegno di eliminazione oggetto scaduto.

In questo esempio viene mostrato uno scenario che può creare contrassegni di eliminazione oggetto scaduto nel bucket. Viene inoltre mostrato come utilizzare la configurazione del ciclo di vita S3 per indicare ad Amazon S3 di rimuovere i contrassegni di eliminazione oggetto scaduto.

Supponiamo di scrivere una configurazione del ciclo di vita S3 che utilizzi l'`NoncurrentVersionExpiration`azione per rimuovere le versioni non correnti 30 giorni dopo che sono diventate non correnti e mantenga al massimo 10 versioni non correnti, come mostrato nell'esempio seguente.

```
<LifecycleConfiguration>
    <Rule>
        ...
        <NoncurrentVersionExpiration>
            <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
            <NoncurrentDays>30</NoncurrentDays>
        </NoncurrentVersionExpiration>
    </Rule>
```

```
</LifecycleConfiguration>
```

L'operazione `NoncurrentVersionExpiration` non si applica alle versioni correnti dell'oggetto. Rimuove solamente le versioni non correnti.

Per le versioni dell'oggetto correnti esistono le seguenti opzioni per gestirne la durata a seconda che le versioni dell'oggetto correnti seguano un ciclo di vita ben definito:

- Le versioni correnti dell'oggetto seguono un ciclo di vita ben definito.

  In questo caso si può utilizzare una configurazione del ciclo di vita S3 con l'operazione `Expiration` per indicare ad Amazon S3 di rimuovere le versioni correnti, come mostrato nell'esempio seguente.

```
<LifecycleConfiguration>
    <Rule>
        ...
        <Expiration>
           <Days>60</Days>
        </Expiration>
        <NoncurrentVersionExpiration>
            <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
            <NoncurrentDays>30</NoncurrentDays>
        </NoncurrentVersionExpiration>
    </Rule>
</LifecycleConfiguration>
```

  Amazon S3 rimuove le versioni correnti 60 giorni dopo la data di creazione aggiungendo un contrassegno di eliminazione per ognuna delle versioni dell'oggetto correnti. La versione corrente diventa quindi non corrente e il contrassegno di eliminazione diventa la versione corrente. Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

  > **Note**
  >
  > Non è possibile specificare sia un tag `Days` che un tag `ExpiredObjectDeleteMarker` sulla stessa regola. Specificando il tag `Days`, Amazon S3 eseguirà automaticamente la pulizia di `ExpiredObjectDeleteMarker` una volta che i contrassegni di eliminazione sono abbastanza vecchi da soddisfare i criteri di età. È possibile creare una regola

separata con solo il tag `ExpiredObjectDeleteMarker` per ripulire i contrassegni di eliminazione non appena diventano l'unica versione.

L'operazione `NoncurrentVersionExpiration` nella stessa configurazione del ciclo di vita S3 rimuove gli oggetti non correnti 30 giorni dopo che sono diventati non correnti. Pertanto, in questo esempio, tutte le versioni degli oggetti vengono rimosse in modo permanente 90 giorni dopo la creazione dell'oggetto. Nonostante i contrassegni di eliminazione degli oggetti scaduti vengano creati durante questo processo, Amazon S3 rileva e rimuove i contrassegni di eliminazione degli oggetti scaduti per te.

- Versioni correnti dell'oggetto che non seguono un ciclo di vita ben definito.

  In questo caso è possibile rimuovere gli oggetti manualmente quando non servono più, creando un contrassegno di eliminazione con una o più versioni non correnti. Se la configurazione del ciclo di vita S3 con l'operazione `NoncurrentVersionExpiration` rimuove tutte le versioni non correnti, rimarranno i contrassegni di eliminazione oggetto scaduto.

  In questo specifico scenario la configurazione del ciclo di vita S3 fornisce un'operazione `Expiration` che puoi utilizzare per rimuovere i contrassegni di eliminazione oggetto scaduto.

```
<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>
```

Se si imposta l'elemento `ExpiredObjectDeleteMarker` su `true` nell'operazione `Expiration`, si indica ad Amazon S3 di rimuovere i contrassegni di eliminazione oggetto scaduto.

> **Note**
>
> Quando si specifica l'operazione del ciclo di vita S3 `ExpiredObjectDeleteMarker`, nella regola non può essere specificato un filtro basato su tag.

Esempio 8: configurazione del ciclo di vita per interrompere i caricamenti in più parti

È possibile utilizzare l'operazione REST API di Amazon S3 per il caricamento in più parti per caricare oggetti di grandi dimensioni in parti. Per ulteriori informazioni sui caricamenti in più parti, consulta la sezione Caricamento e copia di oggetti utilizzando il caricamento in più parti.

Utilizzando una configurazione del ciclo di vita di S3, puoi fare in modo che Amazon S3 interrompa i caricamenti incompleti in più parti (identificati dal prefisso del nome chiave specificato nella regola) se non vengono completati entro un determinato numero di giorni dall'avvio. Quando Amazon S3 interrompe un caricamento in più parti, elimina tutte le parti associate al caricamento in più parti. Questo processo aiuta a controllare i costi di archiviazione garantendo che non siano presenti caricamenti in più parti incompleti con parti archiviate in Amazon S3.

> **Note**
>
> Quando si specifica l'operazione del ciclo di vita S3 `AbortIncompleteMultipartUpload`, nella regola non può essere specificato un filtro basato su tag.

Di seguito è riportata una configurazione del ciclo di vita S3 di esempio che specifica una regola con l'operazione `AbortIncompleteMultipartUpload`. Questa operazione richiede ad Amazon S3 di interrompere i caricamenti in più parti incompleti sette giorni dopo l'avvio.

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Filter>
           <Prefix>SomeKeyPrefix/</Prefix>
        </Filter>
        <Status>rule-status</Status>
```

```
        <AbortIncompleteMultipartUpload>
          <DaysAfterInitiation>7</DaysAfterInitiation>
        </AbortIncompleteMultipartUpload>
    </Rule>
</LifecycleConfiguration>
```

Esempio 9: Configurazione del ciclo di vita con regole basate sulle dimensioni

È possibile creare regole per la transizione degli oggetti in base alle dimensioni. Puoi specificare una dimensione minima (`ObjectSizeGreaterThan`) o una dimensione massima (`ObjectSizeLessThan`) oppure puoi specificare un intervallo di dimensioni dell'oggetto (in byte). Quando si utilizzano più filtri, ad esempio un prefisso e una regola di dimensione, è necessario racchiudere i filtri in un elemento <And>.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Transition with a prefix and based on size</ID>
    <Filter>
       <And>
          <Prefix>tax/</Prefix>
          <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
       </And>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Se stai specificando un intervallo utilizzando entrambi gli elementi `ObjectSizeGreaterThan` e `ObjectSizeLessThan`, la dimensione massima dell'oggetto deve essere maggiore della dimensione minima dell'oggetto. Quando si utilizzano più filtri, è necessario racchiudere i filtri in un elemento <And>. L'esempio seguente mostra come specificare oggetti in un intervallo compreso tra 500 byte e 64.000 byte. Quando si specifica un intervallo, i `ObjectSizeLessThan` filtri `ObjectSizeGreaterThan` and escludono i valori specificati. Per ulteriori informazioni, consulta the section called “Elemento Filter”.

```
<LifecycleConfiguration>
```

```
    <Rule>
        ...
          <And>
             <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
             <ObjectSizeLessThan>64000</ObjectSizeLessThan>
          </And>
    </Rule>
</LifecycleConfiguration>
```

È inoltre possibile creare regole per far scadere in modo specifico gli oggetti non correnti che non contengono dati, inclusi gli oggetti contrassegno di eliminazione non correnti creati in un bucket abilitato al controllo delle versioni. Nell'esempio seguente viene utilizzata l'azione `NoncurrentVersionExpiration` per rimuovere le versioni non correnti 30 giorni dopo che sono diventate non correnti e mantenere al massimo 10 versioni non correnti di oggetti. Inoltre, utilizza l'elemento `ObjectSizeLessThan` per filtrare solo gli oggetti senza dati.

```
<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>
```

Amazon S3 Inventory

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3

Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva negli SDK and. AWS Command Line Interface AWS Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

Puoi usare Inventario Amazon S3 per gestire lo storage. Ad esempio, puoi utilizzarlo per effettuare l'audit e creare report sullo stato di replica e crittografia degli oggetti per attività, conformità ed esigenze normative. Inventario Amazon S3 può anche semplificare e accelerare flussi di lavoro aziendali e processi con Big Data, fornendo un'alternativa pianificata all'operazione API sincrona `List` in Amazon S3. Inventario Amazon S3 non utilizza le operazioni API `List` per l'audit degli oggetti e non influisce sulla velocità di richiesta del bucket.

Inventario Amazon S3 fornisce file di output con valori separati da virgole (CSV), Apache Optimized Row Columnar (ORC) o Apache Parquet che elencano gli oggetti e i metadati corrispondenti con frequenza giornaliera o settimanale per un bucket S3 o oggetto con un prefisso condiviso, ovvero oggetti che hanno nomi che iniziano con una stringa comune. Se si imposta un inventario settimanale, un report viene generato ogni domenica (fuso orario UTC) dopo il report iniziale. Per informazioni sui prezzi di Amazon S3 Inventory, consulta Prezzi di Amazon S3.

È possibile configurare diversi elenchi di inventario per un bucket. Quando si configura un elenco di inventario, è possibile specificare quanto segue:

- Quali metadati degli oggetti includere nell'inventario
- Se elencare tutte le versioni degli oggetti o solo le versioni correnti
- Dove archiviare l'output del file dell'elenco di inventario
- Se generare l'inventario su base giornaliera o settimanale
- Se crittografare il file dell'elenco di inventario

Puoi eseguire query su Inventario Amazon S3 con query SQL standard utilizzando Amazon Athena, Amazon Redshift Spectrum e altri strumenti, come Presto, Apache Hive e Apache Spark. Per ulteriori informazioni sull'uso di Athena per effettuare query sui file di inventario, consulta the section called “Esecuzione di query sull'inventario con Athena”.

Bucket di origine e di destinazione

Il bucket per il quale l'inventario elenca gli oggetti è chiamato bucket di origine. Il bucket nel quale viene archiviato il file dell'elenco di inventario è denominato bucket di destinazione.

Bucket di origine

L'inventario elenca gli oggetti che sono archiviati nel bucket di origine. È possibile ottenere un elenco di inventario per un intero bucket o filtrarlo in base al prefisso del nome della chiave dell'oggetto.

Il bucket di origine:

- Contiene gli oggetti elencati nell'inventario
- Contiene la configurazione per l'inventario

Bucket di destinazione

I file dell'elenco di Amazon S3 Inventory vengono scritti nel bucket di destinazione. Per raggruppare tutti i file di elenco dell'inventario in un percorso comune del bucket di destinazione, puoi specificare un prefisso di destinazione nella configurazione dell'inventario.

Il bucket di destinazione:

- Contiene gli elenchi dei file dell'inventario.
- Contiene i file manifesto che elencano tutti i file dell'inventario che sono archiviati nel bucket di destinazione. Per ulteriori informazioni, consulta Manifest inventario.
- Deve avere una policy del bucket per concedere ad Amazon S3 le autorizzazioni necessarie per verificare la proprietà del bucket e per scrivere file nel bucket.
- Deve essere nello stesso del bucket di origine. Regione AWS
- Può coincidere con il bucket di origine.
- Può essere di proprietà di un account Account AWS diverso da quello che possiede il bucket di origine.

Elenco di Amazon S3 Inventory

Un file dell'elenco di inventario contiene un elenco degli oggetti presenti nel bucket di origine e i metadata per ogni oggetto. Un file dell'elenco di inventario viene archiviato nel bucket di destinazione con uno dei seguenti formati:

- Come un file CSV compresso con GZIP
- Come un file in formato ORC (optimized row columnar) Apache compresso con ZLIB
- Come un file Apache Parquet compresso con Snappy

> **Note**
>
> Non è garantito che gli oggetti nei report di Inventario Amazon S3 siano ordinati in qualsiasi ordine.

Un file dell'elenco di inventario contiene un elenco degli oggetti presenti nel bucket di origine e i metadata per ogni oggetto elencato:

- Bucket Name (Nome bucket): nome del bucket cui è destinato l'inventario.
- Nome chiave: il nome della chiave dell'oggetto (o chiave) che identifica in modo univoco l'oggetto nel bucket. Se si utilizza il formato di file CSV, il nome della chiave è codificato in formato URL e dovrà essere decodificato prima di poterlo utilizzare.
- ID versione: l'ID versione dell'oggetto. Quando viene attivata la funzione Controllo delle versioni in un bucket, Amazon S3 assegna un numero di versione agli oggetti aggiunti al bucket. Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3. (Questo campo non è incluso se l'elenco è configurato unicamente per la versione corrente degli oggetti).
- IsLatest— Imposta `True` se l'oggetto è la versione corrente dell'oggetto. (Questo campo non è incluso se l'elenco è configurato unicamente per la versione corrente degli oggetti).
- Contrassegno di eliminazione: impostato su `True` se l'oggetto è un contrassegno di eliminazione. Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3. (Questo campo viene aggiunto automaticamente al report se questo è stato configurato per comprendere tutte le versioni degli oggetti).
- Dimensione: la dimensione dell'oggetto in byte, esclusa la dimensione dei caricamenti incompleti in più parti, dei metadati degli oggetti e dei contrassegni di eliminazione.
- Data dell'ultima modifica: la più recente tra la data di creazione dell'oggetto o la data dell'ultima modifica.
- ETag: il tag di entità (ETag) è un hash dell'oggetto. L'ETag riflette solo i cambiamenti ai contenuti di un oggetto, non ai suoi metadati. Questo ETag può essere o meno un digest MD5 dei dati oggetto. a seconda di come l'oggetto è stato creato e crittografato.
- Classe di archiviazione: la classe di archiviazione utilizzata per archiviare l'oggetto. impostato su `STANDARD`, `REDUCED_REDUNDANCY`, `STANDARD_IA`, `ONEZONE_IA`, `INTELLIGENT_TIERING`, `GLACIER`, `DEEP_ARCHIVE`, `OUTPOSTS`, `GLACIER_IR` o `SNOW`. Per ulteriori informazioni, consulta Utilizzo delle classi di storage di Amazon S3.

- Multipart upload flag (Contrassegno di caricamento in più parti): impostato su `True` se l'oggetto è stato caricato come caricamento in più parti. Per ulteriori informazioni, consulta Caricamento e copia di oggetti utilizzando il caricamento in più parti.
- Stato della replica: impostare su `PENDING`, `COMPLETED`, `FAILED` oppure`REPLICA`. Per ulteriori informazioni, consulta Ottenimento delle informazioni sullo stato della replica.
- Stato della crittografia: lo stato della crittografia lato server, a seconda del tipo di chiave di crittografia utilizzata: una chiave gestita da Amazon S3 (SSE-S3), una chiave () (SSE-KMS) o AWS KMS una chiave AWS Key Management Service fornita dal cliente (SSE-C). Impostata su `SSE-S3`, `SSE-C`, `SSE-KMS` o `NOT-SSE`. Uno stato di `NOT-SSE` indica che l'oggetto non è crittografato con crittografia lato server. Per ulteriori informazioni, consulta Protezione dei dati con la crittografia.
- Data di fine conservazione del blocco oggetti S3: data fino alla quale l'oggetto bloccato non può essere eliminato. Per ulteriori informazioni, consulta Utilizzo del blocco oggetti S3.
- Modalità di conservazione del blocco oggetti S3: impostata su `Governance` o `Compliance` per gli oggetti bloccati. Per ulteriori informazioni, consulta Utilizzo del blocco oggetti S3.
- Stato legale blocco oggetti S3: impostato su `On` se è stato applicato un blocco di carattere legale a un oggetto, altrimenti su un oggetto. Altrimenti il valore è impostato su `Off`. Per ulteriori informazioni, consulta Utilizzo del blocco oggetti S3.
- Livello di accesso S3 Intelligent-Tiering: livello di accesso (frequente o raro) dell'oggetto se archiviato nella classe di archiviazione S3 Intelligent-Tiering. Impostata su `FREQUENT`, `INFREQUENT`, `ARCHIVE_INSTANT_ACCESS`, `ARCHIVE` o `DEEP_ARCHIVE`. Per ulteriori informazioni, consulta Classe di storage per ottimizzare automaticamente i dati con modelli di accesso variabili o sconosciuti.
- Stato delle chiavi bucket S3: impostare su `ENABLED` o `DISABLED`. Indica se l'oggetto utilizza la chiave bucket S3 per SSE-KMS. Per ulteriori informazioni, consulta Utilizzo di chiavi bucket Amazon S3.
- Algoritmo di checksum: indica l'algoritmo usato per creare il checksum dell'oggetto.
- Elenco di controllo degli accessi agli oggetti: una lista di controllo degli accessi (ACL) per ogni oggetto che definisce a quali Account AWS o gruppi è concesso l'accesso a tale oggetto e il tipo di accesso concesso. Il campo ACL oggetto è definito in formato JSON. Un report Inventario S3 include gli ACL associati agli oggetti nel bucket di origine, anche quando gli ACL sono disabilitati per il bucket. Per ulteriori informazioni, consultare Utilizzo del campo ACL oggetto e Panoramica delle liste di controllo accessi (ACL).

> **Note**
>
> Il campo ACL oggetto è definito in formato JSON. Un report di inventario visualizza il valore per il campo ACL oggetto come stringa con codifica base64.
> Ad esempio, si supponga di avere il seguente campo ACL oggetto in formato JSON:
>
> ```
> {
>         "version": "2022-11-10",
>         "status": "AVAILABLE",
>         "grants": [{
>             "canonicalId": "example-canonical-user-ID",
>             "type": "CanonicalUser",
>             "permission": "READ"
>         }]
> }
> ```
>
> Il campo ACL oggetto è codificato e visualizzato come la seguente stringa con codifica base64:
>
> ```
> eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3siY2Fub25pY2Fs
> ```
>
> Per ottenere il valore decodificato in JSON per il campo ACL oggetto, puoi eseguire una query su questo campo in Amazon Athena. Per ulteriori esempi di query, consulta Esecuzione di query sull'inventario Amazon S3 con Amazon Athena.

- Proprietario dell'oggetto: il proprietario dell'oggetto.

> **Note**
>
> Quando un oggetto raggiunge la fine del suo ciclo di vita in base alla relativa configurazione, Amazon S3 lo aggiunge alla coda degli oggetti da eliminare e lo rimuove in modo asincrono. Deve pertanto esistere un ritardo tra la data di scadenza dell'oggetto e la data in cui Amazon S3 rimuove tale oggetto. Il report di inventario include gli oggetti scaduti ma non ancora rimossi. Per ulteriori informazioni sulle operazioni di scadenza nel ciclo di vita S3, consulta Oggetti in scadenza.

Consigliamo di creare una policy del ciclo di vita che elimini i vecchi elenchi di inventario. Per ulteriori informazioni, consulta Gestione del ciclo di vita dello storage.

L'autorizzazione `s3:PutInventoryConfiguration` consente all'utente di selezionare tutti i campi di metadati elencati in precedenza per ogni oggetto durante la configurazione di un elenco inventario e di specificare il bucket di destinazione in cui archiviare l'inventario. Un utente con accesso in lettura agli oggetti nel bucket di destinazione può accedere a tutti i campi di metadati degli oggetti disponibili nell'elenco inventario. Per limitare l'accesso a un report di inventario, consulta Concedere autorizzazioni per S3 Inventory e S3 Analytics.

Consistenza dell'inventario

In ogni elenco di inventario potrebbero non comparire tutti gli oggetti. L'elenco inventario fornisce l'eventuale consistenza delle richieste PUT (sia di nuovi oggetti che di sovrascritture) e delle richieste `DELETE`. Ogni elenco di inventario per un bucket è una snapshot degli elementi del bucket. Questi elenchi sono alla fine coerenti (ovvero, un elenco potrebbe non includere oggetti aggiunti o eliminati di recente).

Per convalidare lo stato di un oggetto prima di agire sull'oggetto stesso, consigliamo di effettuare una richiesta REST API `HeadObject` per recuperare i metadati dell'oggetto o controllare le proprietà dell'oggetto nella console di Amazon S3. Puoi anche controllare i metadati degli oggetti con AWS CLI o con gli AWS SDK. Per ulteriori informazioni, consulta `HeadObject` in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Per ulteriori informazioni sull'utilizzo di Amazon S3 Inventory, consulta gli argomenti riportati di seguito.

Argomenti

- Configurazione di Amazon S3 Inventory
- Impostazione delle notifiche di eventi Amazon S3 per il completamento dell'inventario
- Individuazione dell'elenco inventario
- Esecuzione di query sull'inventario Amazon S3 con Amazon Athena
- Convertire stringhe di ID versione vuote nei report Inventario Amazon S3 in stringhe nulle
- Utilizzo del campo ACL oggetto

Configurazione di Amazon S3 Inventory

Amazon S3 Inventory fornisce un elenco di tipo file flat contenente oggetti e metadati in base a una pianificazione definita. Puoi utilizzare S3 Inventory come alternativa pianificata all'operazione API sincrona `List` di Amazon S3. S3 Inventory fornisce file di output con valori separati da virgole (CSV), in Apacheformato ORC (Optimized Row Columnar) o in formato Apache Parquet (Parquet) che elencano gli oggetti e i metadati corrispondenti.

Puoi configurare S3 Inventory per creare elenchi di inventario su base giornaliera o settimanale per un bucket S3 o per oggetti che condividono un prefisso (oggetti con nomi che iniziano con la stessa stringa). Per ulteriori informazioni, consulta Amazon S3 Inventory.

In questa sezione viene descritto come configurare un inventario inserendo le informazioni sui bucket di origine e destinazione dell'inventario.

Argomenti

Panoramica

Amazon S3 Inventory semplifica la gestione dell'archiviazione tramite la creazione di elenchi di oggetti in un bucket S3 in base a una pianificazione definita. È possibile configurare diversi elenchi di inventario per un bucket. Gli elenchi di inventario sono pubblicati su file CSV, ORC o Parquet in un bucket di destinazione.

Il modo più semplice per configurare un inventario è utilizzare la console Amazon S3, ma puoi anche utilizzare l'API REST di Amazon S3 AWS Command Line Interface ,AWS CLI() o gli SDK. AWS La console effettua la prima fase della seguente procedura: l'aggiunta di una policy di bucket al bucket di destinazione.

Per configurare un Amazon S3 Inventory per un bucket S3

1. Aggiungere una policy di bucket per il bucket di destinazione.

   È necessario creare una policy sul bucket di destinazione che conceda le autorizzazioni ad Amazon S3 per scrivere oggetti nel bucket nella posizione definita. Per un esempio di policy, consulta Concedere autorizzazioni per S3 Inventory e S3 Analytics.

2. Configurare un inventario per elencare gli oggetti in un bucket di origine e pubblicare l'elenco su un bucket di destinazione.

   Quando si configura un elenco di inventario per un bucket di origine, viene specificato il bucket di destinazione dove si intende archiviare l'elenco, indicando se si vuole generare l'elenco giornalmente o settimanalmente. Puoi anche configurare se elencare tutte le versioni degli oggetti o solo le versioni correnti e quali metadati degli oggetti includere.

   Alcuni campi di metadati degli oggetti nelle configurazioni dei report di S3 Inventory sono facoltativi, il che significa che sono disponibili per impostazione predefinita, ma possono essere limitati quando si concede l'autorizzazione a un utente. `s3:PutInventoryConfiguration` Puoi controllare se gli utenti possono includere questi campi di metadati opzionali nei loro report utilizzando la chiave di condizione. `s3:InventoryAccessibleOptionalFields`

   Per ulteriori informazioni sui campi di metadati opzionali disponibili in S3 Inventory, consulta OptionalFieldsAmazon Simple Storage Service API Reference. Per ulteriori informazioni sulla limitazione dell'accesso a determinati campi di metadati opzionali in una configurazione di inventario, consulta. Controlla la creazione della configurazione dei report di S3 Inventory

   Puoi specificare che il file della lista di inventario sia crittografato utilizzando la crittografia lato server con una chiave gestita Amazon S3 (SSE-S3) o AWS Key Management Service una () chiave gestita dal cliente AWS KMS(SSE-KMS).

   > **Note**
   >
   > Il Chiave gestita da AWS (`aws/s3`) non è supportato per la crittografia SSE-KMS con S3 Inventory.

   Per ulteriori informazioni su SSE-S3 e SSE-KMS, consulta Protezione dei dati con la crittografia lato server. Se si intende utilizzare la crittografia SSE-KMS, consulta la Fase 3.

   - Per informazioni su come utilizzare la console per configurare un elenco inventario, consulta Configurazione dell'inventario utilizzando la console S3.

- Per utilizzare l'API Amazon S3 per configurare un elenco di inventario, utilizza l'operazione API PutBucketInventoryConfigurationREST o l'equivalente degli SDK AWS CLI o AWS .

3. Per crittografare il file dell'elenco di inventario con SSE-KMS, concedi a Simple Storage Service (Amazon S3) l'autorizzazione per l'utilizzo della AWS KMS key.

   Puoi configurare la crittografia per il file dell'elenco di inventario utilizzando la console Amazon S3, l'API REST AWS CLI di Amazon S3 o gli SDK. AWS Indipendentemente dalla soluzione scelta, devi concedere ad Amazon S3 l'autorizzazione per l'utilizzo della chiave gestita dal cliente per crittografare il file di inventario. Per concedere ad Amazon S3 l'autorizzazione, modifica la policy della chiave gestita dal cliente che desideri utilizzare per crittografare il file di inventario. Per ulteriori informazioni, consulta Concessione ad Amazon S3 dell'autorizzazione per l'utilizzo della chiave gestita dal cliente per la crittografia.

   Il bucket di destinazione in cui è archiviato il file dell'elenco di inventario può essere di proprietà di un Account AWS diverso rispetto all'account che possiede il bucket di origine. Se utilizzi la crittografia SSE-KMS per le operazioni tra account di Amazon S3 Inventory, ti consigliamo di utilizzare una chiave ARN KMS completamente qualificata quando configuri l'inventario S3. Per ulteriori informazioni, consulta Utilizzo della crittografia SSE-KMS per operazioni multi-account e ServerSideEncryptionByDefault nella Documentazione di riferimento delle API Amazon Simple Storage Service.

Creazione di una policy di bucket di destinazione

Se crei la configurazione dell'inventario tramite la console Amazon S3, Amazon S3 crea automaticamente una policy di bucket sul bucket di destinazione che concede ad Amazon S3 l'autorizzazione di scrittura. Tuttavia, se crei la configurazione dell'inventario tramite gli AWS CLI AWS SDK o l'API REST di Amazon S3, devi aggiungere manualmente una bucket policy sul bucket di destinazione. Per ulteriori informazioni, consulta Concedere autorizzazioni per S3 Inventory e S3 Analytics. La policy S3 Inventory destination bucket consente ad Amazon S3 di scrivere i dati per i report di inventario nel bucket.

Se si verifica un errore quando si tenta di creare la policy del bucket, vengono fornite le istruzioni su come correggerlo. Ad esempio, se scegli un bucket di destinazione in un altro Account AWS e non disponi delle autorizzazioni per leggere e scrivere nella policy del bucket, viene visualizzato un messaggio di errore.

In questo caso, il proprietario del bucket di destinazione deve aggiungere la policy del bucket al bucket di destinazione. Se la policy non viene aggiunta al bucket di destinazione, non si otterrà alcun

report di inventario, in quanto Amazon S3 non dispone dell'autorizzazione di scrittura per il bucket di destinazione. Se il bucket di origine è di proprietà di un account diverso da quello dell'utente attuale, l'ID account corretto del proprietario del bucket di origine verrà sostituito nella policy.

Concessione ad Amazon S3 dell'autorizzazione per l'utilizzo della chiave gestita dal cliente per la crittografia

Per concedere ad Amazon S3 l'autorizzazione a utilizzare la tua chiave gestita dal cliente AWS Key Management Service (AWS KMS) per la crittografia lato server, devi utilizzare una policy di chiave. Per aggiornare la policy della chiave in modo da poter utilizzare la chiave gestita dal cliente, completa la procedura seguente.

Per concedere ad Amazon S3 le autorizzazioni di crittografia utilizzando la chiave gestita dal cliente

1. Utilizzando la Account AWS chiave proprietaria della chiave gestita dal cliente, accedi a. AWS Management Console
2. Apri la AWS KMS console all'indirizzo https://console.aws.amazon.com/kms.
3. Per modificare la Regione AWS, usa il selettore della regione nell'angolo superiore destro della pagina.
4. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
5. In Chiavi gestite dal cliente, scegli la chiave gestita dal cliente che desideri utilizzare per crittografare i file inventario.
6. Nella sezione Key Policy (Policy chiave), scegliere Switch to policy view (Passa alla visualizzazione della policy).
7. Per aggiornare la policy chiave, seleziona Modifica.
8. Nella pagina Modifica policy delle chiavi, aggiungi le seguenti righe alla policy della chiave esistente. Per *source-account-id* e *example-s3-source-bucket*, fornisci i valori appropriati per il tuo caso d'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:::example-s3-source-bucket"
     }
   }
}
```

9. Seleziona Salvataggio delle modifiche.

Per ulteriori informazioni sulla creazione di chiavi gestite dal cliente e sull'utilizzo delle policy delle chiavi, consulta i seguenti collegamenti nella Guida per Developer di AWS Key Management Service :

- Gestione delle chiavi
- Politiche chiave in AWS KMS

Configurazione dell'inventario utilizzando la console S3

Segui queste istruzioni per configurare l'inventario utilizzando la console S3.

> **Note**
>
> La consegna del primo report di inventario Amazon S3 può richiedere fino a 48 ore.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket). Nell'elenco Bucket seleziona il nome del bucket per cui desideri configurare Inventario Amazon S3.
3. Scegliere la scheda Management (Gestione),
4. In Configurazioni inventario seleziona Crea configurazione inventario.
5. Specifica un nome in Nome della configurazione dell'inventario.
6. Per Ambito dell'inventario, esegui le operazioni descritte di seguito.

- Immetti un prefisso facoltativo.
- Scegli quali versioni dell'oggetto includere, Solo versioni correnti o Includi tutte le versioni.

7. In Dettagli report scegli la posizione dell'account Account AWS in cui desideri salvare i report: Questo account o Un account diverso.
8. In Destinazione, scegli il bucket di destinazione in cui desideri salvare i report di inventario.

   Il bucket di destinazione deve trovarsi nello Regione AWS stesso bucket per il quale stai configurando l'inventario. Il bucket di destinazione può trovarsi in un diverso Account AWS. Quando specifichi il bucket di destinazione, puoi anche includere un prefisso opzionale per raggruppare insiemi i report di inventario.

   Nel campo Bucket di destinazione viene visualizzata l'istruzione Autorizzazione del bucket di destinazione che viene aggiunta alla policy del bucket di destinazione per consentire ad Amazon S3 di inserire i dati in tale bucket. Per ulteriori informazioni, consulta Creazione di una policy di bucket di destinazione.
9. In Frequenza, seleziona la frequenza con cui verrà generato il report: Giornaliero o Settimanale.
10. Per Formato di output, scegli uno dei seguenti formati per il report:

    - CSV: se prevedi di utilizzare questo report di inventario con Operazioni in batch S3 o se desideri analizzare questo report in un altro strumento, come Microsoft Excel, scegli CSV.
    - Apache ORC
    - Apache Parquet
11. In Stato seleziona Abilita o Disabilita.
12. Per configurare la crittografia lato server, in Crittografia dei report di inventario, segui la procedura riportata sotto:

    a. In Crittografia lato server, scegli Non specificare una chiave di crittografia o Specificare una chiave di crittografia per crittografare i dati.

       - Per conservare le impostazioni relative ai bucket per la crittografia predefinita degli oggetti lato server durante l'archiviazione in Amazon S3, scegli Non specificare una chiave di crittografia. Finché nella destinazione del bucket sono abilitate le chiavi bucket S3, l'operazione di copia applica la chiave bucket S3 al bucket di destinazione.

> **Note**
>
> Se la policy del bucket per la destinazione specificata richiede la crittografia degli oggetti prima di archiviarli in Amazon S3, devi scegliere Specificare una chiave di crittografia. In caso contrario, la copia degli oggetti nella destinazione avrà esito negativo.

- Per crittografare gli oggetti prima di archiviarli in Amazon S3, scegli Specifica una chiave di crittografia.

b. Se hai scelto Specificare una chiave di crittografia, in Tipo di crittografia, devi scegliere una chiave gestita Amazon S3 (SSE-S3) o AWS Key Management Service una chiave (SSE-KMS).

   Per crittografare gli oggetti, SSE-S3 utilizza una delle cifrature di blocco più complesse, lo standard di crittografia avanzata a 256 bit (AES-256). SSE-KMS garantisce un maggiore controllo sulla chiave. Per ulteriori informazioni su SSE-S3, consulta Uso della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3). Per ulteriori informazioni su SSE-KMS, consulta Utilizzo della crittografia lato server con chiavi (SSE-KMS) AWS KMS.

   > **Note**
   >
   > Per crittografare il file elenco inventario con SSE-KMS, devi impostare Amazon S3 in modo che possa utilizzare la chiave gestita dal cliente. Per le istruzioni, consulta la sezione Concedere ad Amazon S3 l'autorizzazione delle chiavi KMS per la crittografia.

c. Se hai scelto AWS Key Management Service la chiave (SSE-KMS), sotto AWS KMS key, puoi specificare la tua chiave tramite una delle seguenti opzioni. AWS KMS

   > **Note**
   >
   > Se il bucket di destinazione che memorizza il file dell'elenco di inventario è di proprietà di un altro Account AWS, assicurati di utilizzare una chiave KMS ARN completa per specificare la tua chiave KMS.

- Per scegliere da un elenco di chiavi KMS disponibili, scegli tra le tue AWS KMS chiavi e scegli una chiave KMS con crittografia simmetrica dall'elenco delle chiavi disponibili. Assicurati che la chiave KMS si trovi nella stessa regione del tuo bucket.

  > **Note**
  >
  > Nell'elenco vengono visualizzate sia la chiave Chiave gestita da AWS (`aws/s3`) che quella gestita dai clienti. Tuttavia, Chiave gestita da AWS (`aws/s3`) non è supportato per la crittografia SSE-KMS con S3 Inventory.

- Per inserire l'ARN della chiave KMS, scegli Inserisci la AWS KMS chiave ARN e inserisci l'ARN della chiave KMS nel campo visualizzato.
- Per creare una nuova chiave gestita dal cliente nella AWS KMS console, scegli Crea una chiave KMS.

13. Per Campi di metadati aggiuntivi, seleziona una o più delle seguenti opzioni per aggiungere il report di inventario:

    - Dimensione: la dimensione dell'oggetto in byte, esclusa la dimensione dei caricamenti incompleti in più parti, dei metadati degli oggetti e dei contrassegni di eliminazione.
    - Data dell'ultima modifica: la più recente tra la data di creazione dell'oggetto o la data dell'ultima modifica.
    - Caricamento in più parti: specifica che l'oggetto è stato caricato in più parti. Per ulteriori informazioni, consulta Caricamento e copia di oggetti utilizzando il caricamento in più parti.
    - Stato di replica: lo stato di replica dell'oggetto. Per ulteriori informazioni, consulta Ottenimento delle informazioni sullo stato della replica.
    - Stato crittografia: il tipo di crittografia lato server utilizzata per crittografare l'oggetto. Per ulteriori informazioni, consulta Protezione dei dati con la crittografia lato server.
    - Stato della chiave del bucket: indica se una chiave a livello di bucket generata da AWS KMS si applica all'oggetto. Per ulteriori informazioni, consulta Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3.
    - Elenco di controllo dell'accesso agli oggetti: una lista di controllo degli accessi (ACL) per ogni oggetto che definisce a quali Account AWS o gruppi è concesso l'accesso a questo oggetto e il tipo di accesso concesso. Per ulteriori informazioni su questo campo, consulta Utilizzo del

campo ACL oggetto. Per ulteriori informazioni sulle ACL, consulta Panoramica delle liste di controllo accessi (ACL).

- Proprietario dell'oggetto: il proprietario dell'oggetto.
- Classe di archiviazione: la classe di archiviazione utilizzata per archiviare l'oggetto.
- Intelligent-Tiering: livello di accesso: indica il livello di accesso (frequente o raro) dell'oggetto se è stato archiviato nella classe di archiviazione S3 Intelligent-Tiering. Per ulteriori informazioni, consulta Classe di storage per ottimizzare automaticamente i dati con modelli di accesso variabili o sconosciuti.
- ETag: il tag di entità (ETag) è un hash dell'oggetto. L'ETag riflette solo i cambiamenti ai contenuti di un oggetto, non ai suoi metadati. L'ETag potrebbe o meno essere un digest MD5 dei dati dell'oggetto. a seconda di come l'oggetto è stato creato e crittografato. Per ulteriori informazioni, consulta Object in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).
- Algoritmo di checksum: indica l'algoritmo usato per creare il checksum dell'oggetto.
- Tutte le configurazioni di blocco degli oggetti: lo stato di blocco dell'oggetto, incluse le seguenti impostazioni:
  - Blocco degli oggetti: modalità di conservazione: il livello di protezione applicato all'oggetto, Governance o Conformità.
  - Blocco degli oggetti: mantenere fino alla data: data fino alla quale l'oggetto bloccato non può essere eliminato.
  - Blocco degli oggetti: status della conservazione di carattere legale: lo stato di conservazione ai fini legali dell'oggetto bloccato.

  Per ulteriori informazioni sul blocco degli oggetti S3, consulta Come funziona il blocco oggetti S3.

Per ulteriori informazioni sul contenuto di un report di inventario, consulta Elenco di Amazon S3 Inventory.

Per ulteriori informazioni sulla limitazione dell'accesso a determinati campi di metadati opzionali in una configurazione di inventario, vedere. Controlla la creazione della configurazione dei report di S3 Inventory

14. Scegli Crea.

Quando viene pubblicato un elenco di inventario, puoi interrogare il file di elenco inventario con Amazon S3 Select. Per ulteriori informazioni su come individuare l'elenco di inventario e interrogare il file dell'elenco di inventario con Amazon S3 Select, consulta Individuazione dell'elenco inventario.

Utilizzo di REST API per utilizzare Inventario S3

Di seguito sono elencate le operazioni REST che puoi utilizzare per lavorare con Amazon S3 Inventory.

- DeleteBucketInventoryConfiguration
- GetBucketInventoryConfiguration
- ListBucketInventoryConfigurations
- PutBucketInventoryConfiguration

Impostazione delle notifiche di eventi Amazon S3 per il completamento dell'inventario

Puoi configurare una notifica di eventi Amazon S3 per ricevere una notifica quando viene creato il file checksum manifest, che indica che è stato aggiunto un elenco di inventario al bucket di destinazione. Il manifesto è un up-to-date elenco di tutti gli elenchi di inventario nella posizione di destinazione.

Amazon S3 può pubblicare eventi in un argomento Amazon Simple Notification Service (Amazon SNS), una coda Amazon Simple Queue Service (Amazon SQS) o una funzione AWS Lambda . Per ulteriori informazioni, consulta Notifiche di eventi Amazon S3.

La seguente configurazione di notifica definisce che tutti i file `manifest.checksum` recentemente aggiunti al bucket di destinazione sono elaborati da AWS Lambda `cloud-function-list-write`.

```
<NotificationConfiguration>
  <QueueConfiguration>
      <Id>1</Id>
      <Filter>
          <S3Key>
              <FilterRule>
                  <Name>prefix</Name>
                  <Value>destination-prefix/source-bucket</Value>
              </FilterRule>
              <FilterRule>
```

```
                <Name>suffix</Name>
                <Value>checksum</Value>
            </FilterRule>
        </S3Key>
    </Filter>
    <Cloudcode>arn:aws:lambda:us-west-2:222233334444:cloud-function-list-write</
Cloudcode>
    <Event>s3:ObjectCreated:*</Event>
  </QueueConfiguration>
  </NotificationConfiguration>
```

Per ulteriori informazioni, consulta Using AWS Lambda with Amazon S3 nella AWS Lambda Developer Guide.

Individuazione dell'elenco inventario

Quando viene pubblicato un elenco di inventario, i file manifest vengono pubblicati nel seguente percorso del bucket di destinazione.

```
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* è il prefisso del nome della chiave dell'oggetto specificato facoltativamente nella configurazione dell'inventario. Puoi utilizzare questo prefisso per raggruppare tutti i file dell'elenco di inventario in un percorso comune all'interno del bucket di destinazione.
- *source-bucket* è il bucket di origine per l'elenco inventario. Il nome del bucket di origine viene aggiunto per evitare conflitti quando più report di inventario provenienti da bucket di origine diversi vengono inviati allo stesso bucket di destinazione.
- *config-ID* viene aggiunto per evitare conflitti con più report di inventario provenienti dallo stesso bucket di origine inviati allo stesso bucket di destinazione. *config-ID* proviene dalla configurazione del report di inventario ed è il nome del report definito durante la configurazione.
- *YYYY-MM-DDTHH-MMZ* è il timestamp composto dall'ora di inizio e dalla data in cui la generazione del report di inventario comincia la scansione del bucket; ad esempio, `2016-11-06T21-32Z`.
- `manifest.json` è il file manifest.
- `manifest.checksum` è l'hash MD5 del contenuto del file `manifest.json`.
- `symlink.txt` è il file manifesto compatibile con Apache Hive.

Gli elenchi di inventario vengono pubblicati giornalmente o settimanalmente nel seguente percorso del bucket di destinazione.

```
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* è il prefisso del nome della chiave dell'oggetto specificato facoltativamente nella configurazione dell'inventario. Puoi utilizzare questo prefisso per raggruppare tutti i file dell'elenco di inventario in un percorso comune nel bucket di destinazione.
- *source-bucket* è il bucket di origine per l'elenco inventario. Il nome del bucket di origine viene aggiunto per evitare conflitti quando più report di inventario provenienti da bucket di origine diversi vengono inviati allo stesso bucket di destinazione.
- *example-file-name*`.csv.gz` è uno dei file CSV di inventario. I nomi di inventario ORC terminano con l'estensione del nome di file `.orc`, mentre i nomi di inventario Parquet terminano con l'estensione del nome di file `.parquet`.

  Puoi richiedere un file di elenco di inventario con Amazon S3 Select. *Nella console Amazon S3, scegli il nome dell'elenco di inventario (ad esempio, destination-prefix/source-bucket/config-ID /data/ .csv.gz). example-file-name* Quindi, scegli Azioni oggetto e Query con S3 Select. Per un esempio di come utilizzare una funzione aggregata S3 Select per interrogare un file di elenco di inventario, vedi SUM Esempio

Manifest inventario

Nei file manifest `manifest.json` e `symlink.txt` viene descritto dove sono posizionati i file di inventario. Ogni volta che viene distribuito un nuovo elenco di inventario, quest'ultimo è accompagnato da un nuovo set di file manifest. Questi file potrebbero sovrascriversi l'un l'altro. Nei bucket con il controllo delle versioni abilitato, Amazon S3 crea nuove versioni dei file manifesto.

Ogni manifesto contenuto nel file `manifest.json` fornisce i metadata e altre informazioni di base riguardanti un inventario. Queste informazioni comprendono:

- Il nome del bucket di origine
- Il nome del bucket di destinazione
- La versione dell'inventario

- La creazione del timestamp in formato data epoca (Unix epoch) che è composto dall'ora di inizio e dalla data in cui il processo di generazione del report di inventario comincia la scansione del bucket
- Il formato e lo schema dei file di inventario
- Un elenco dei file di inventario che si trovano nel bucket di destinazione

Ogni volta che viene scritto un file `manifest.json`, questo è accompagnato da un file `manifest.checksum` che è l'hash MD5 del contenuto del file `manifest.json`.

Example Manifest inventario in un file **`manifest.json`**

Negli esempi seguenti viene illustrato un manifesto inventario in un file `manifest.json` per gli inventari in formato CSV, ORC e Parquet.

CSV

Di seguito viene riportato un esempio di un manifest in un file `manifest.json` per un inventario in 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

Di seguito viene riportato un esempio di un manifest in un file `manifest.json` per un inventario in 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

Di seguito viene riportato un esempio di un manifest in un file `manifest.json` per un inventario in 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"
        }
    ]
}
```

Il file `symlink.txt` è un file manifesto compatibile con Apache Hive che consente a Hive di scoprire automaticamente i file di inventario e i relativi file di dati. Il manifest compatibile con Hive funziona con i servizi compatibili con Hive Athena e Amazon Redshift Spectrum. Funziona anche con applicazioni compatibili con Hive, incluse Presto, Apache Hive, Apache Spark e molte altre.

> **Important**
>
> Il file manifesto compatibile con `symlink.txt` Apache Hive attualmente non può essere utilizzato con AWS Glue.
> La lettura del file `symlink.txt` con Apache Hive e Apache Spark non è supportata per i file di inventario in formato ORC e Parquet.

Esecuzione di query sull'inventario Amazon S3 con Amazon Athena

Puoi eseguire una query sui file di Inventario Amazon S3 con query SQL standard utilizzando Amazon Athena in tutte le regioni in cui Athena è disponibile. Per verificare la disponibilità delle Regione AWS, consulta la Tabella delle Regione AWS.

Athena può eseguire query sui file di Inventario Amazon S3 in formato ORC (optimized row columnar) Apache, Apache Parquet o in formato CSV. Quando si utilizza Athena per eseguire query sui file dell'inventario, è consigliabile utilizzare file di inventario in formato ORC o Parquet. I formati ORC e Parquet consentono di eseguire query più rapidamente e a costi inferiori. ORC e Parquet sono formati di file a colonne autodescrittivi orientati ai tipi progettati per Apache Hadoop. Il formato colonnare permette al lettore di leggere, decomprimere ed elaborare solo le colonne necessarie per la query in corso. I formati ORC e Parquet per Inventario Amazon S3 sono disponibili in tutte le Regioni AWS.

Come utilizzare Athena per eseguire query sui file di Inventario Amazon S3

1. Creare una tabella Athena. Per informazioni sulla creazione di una tabella, consultare Creazione di tabelle in Amazon Athena nella Guida per l'utente di Amazon Athena.
2. Crea la query utilizzando uno dei seguenti modelli di query di esempio, a seconda che esegui la query su un report di inventario in formato ORC, Parquet o CSV.

   - Quando utilizzi Athena per eseguire query su un report di inventario in formato ORC, utilizza la seguente query di esempio come un modello.

     La seguente query di esempio comprende tutti i campi opzionali in un report di inventario in formato ORC.

     Per utilizzare questa query di esempio, effettua le seguenti operazioni:

     - Sostituisci *`your_table_name`* con il nome della tabella Athena creata.
     - Rimuovi gli eventuali campi opzionali che non hai scelto per l'inventario in modo che la query corrisponda ai campi scelti.
     - Sostituisci il nome del bucket e la posizione dell'inventario seguenti (l'ID configurazione) come appropriato in base alla configurazione.

       `s3://`*`DOC-EXAMPLE-BUCKET`*`/`*`config-ID`*`/hive/`

     - Sostituisci la data *`2022-01-01-00-00`* in `projection.dt.range` con il primo giorno dell'intervallo di tempo entro il quale esegui la partizione dei dati in Athena. Per ulteriori informazioni, consulta Partizionamento dei dati in 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"
  );
```

- Quando utilizzi Athena per eseguire query su un report di inventario in formato Parquet, utilizza la seguente query di esempio come un report in formato OCR. Tuttavia, utilizza il seguente Parquet SerDe al posto dell'ORC SerDe nell'istruzione `ROW FORMAT SERDE`.

  ```
  ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
  ```

- Quando utilizzi Athena per eseguire query su un report di inventario in formato CSV, utilizza la seguente query di esempio come un modello.

  La seguente query di esempio comprende tutti i campi opzionali in un report di inventario in formato CSV.

  Per utilizzare questa query di esempio, effettua le seguenti operazioni:

  - Sostituisci *`your_table_name`* con il nome della tabella Athena creata.
  - Rimuovi gli eventuali campi opzionali che non hai scelto per l'inventario in modo che la query corrisponda ai campi scelti.
  - Sostituisci il nome del bucket e la posizione dell'inventario seguenti (l'ID configurazione) come appropriato in base alla configurazione.

s3://*DOC-EXAMPLE-BUCKET*/*config-ID*/hive/

- Sostituisci la data *2022-01-01-00-00* in `projection.dt.range` con il primo giorno dell'intervallo di tempo entro il quale esegui la partizione dei dati in Athena. Per ulteriori informazioni, consulta Partizionamento dei dati in 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. Ora puoi eseguire diverse query sull'inventario, come illustrato negli esempi seguenti. Sostituisci ogni *user input placeholder* con le tue informazioni.

```
# 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;
```

Quando configuri S3 Inventory per aggiungere il campo Elenco di controllo di accesso dell'oggetto (ACL) a un report di inventario, il report visualizza il valore per il campo ACL oggetto come una stringa con codifica base64. Per ottenere il valore decodificato in JSON per il campo ACL oggetto, puoi eseguire una query su questo campo utilizzando Athena. Fare riferimento agli esempi di query riportati di seguito. Per ulteriori informazioni sul campo ACL oggetto, consulta Utilizzo del campo ACL oggetto.

```
# 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;
```

Per ulteriori informazioni sull'utilizzo di Athena, consulta la Guida per l'utente di Amazon Athena.

Convertire stringhe di ID versione vuote nei report Inventario Amazon S3 in stringhe nulle

> **Note**
>
> La procedura seguente si applica solo ai report di Amazon S3 Inventory che includono tutte le versioni e solo se i report "tutte le versioni" vengono utilizzati come manifest per S3 Batch Operations su bucket che hanno il Controllo versioni S3 abilitato. Inoltre, non è necessario convertire stringhe per i report di inventario S3 che specificano solo la versione corrente.

Puoi utilizzare i report di S3 Inventory come manifest per S3 Batch Operations. Tuttavia, quando Controllo versioni S3 è abilitato su un bucket, i report di S3 Inventory che includono tutte le versioni contrassegnano gli oggetti con versione nulla con stringhe vuote nel campo ID versione. Quando un report di Inventory include tutti gli ID versione degli oggetti, Batch Operations riconosce le stringhe `null` come ID di versione ma non le stringhe vuote.

Quando un processo di S3 Batch Operations utilizza come manifest un report "tutte le versioni" di S3 Inventory, non riuscirà a portare a termine tutte le attività sugli oggetti con una stringa vuota nel campo ID versione. Per convertire stringhe vuote nel campo ID versione del report S3 Inventory in stringhe `null` per Batch Operations, attenersi alla procedura seguente.

Aggiorna un report di Amazon S3 Inventory da utilizzare con Batch Operations

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Vai al report di Inventario S3. Il report dell'inventario si trova nel bucket di destinazione specificato durante la configurazione del report dell'inventario. Per ulteriori informazioni sull'individuazione dei report di inventario, consulta Individuazione dell'elenco inventario.
   a. Scegliere il nome del bucket di destinazione.
   b. Scegliere la cartella . La cartella prende il nome dal bucket della fonte d'origine.
   c. Scegli la cartella che prende il nome dalla configurazione di inventario.
   d. Seleziona la casella di spunta accanto alla cartella denominata hive. Nella parte superiore della pagina, scegli Copia URI S3 per copiare l'URI S3 per la cartella.
3. Aprire la console Amazon Athena all'indirizzo https://console.aws.amazon.com/athena/.
4. Nell'editor di query, scegli Impostazioni, quindi seleziona Gestisci. Alla pagina Gestisci impostazioni per Posizione del risultato della query, scegli un bucket S3 in cui archiviare i risultati della query.
5. Nell'editor di query, creare una tabella Athena per conservare i dati nel report di inventario utilizzando il seguente comando. Sostituisci *`table_name`* con un nome a scelta e nella clausola `LOCATION`, inserisci l'URI S3 copiato in precedenza. Quindi scegli Esegui per eseguire la query.

```
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. Per cancellare l'editor di query, scegli Cancella. Quindi, caricare il report di inventario nella tabella utilizzando il comando seguente. Sostituisci il codice *`table_name`* con quello che hai scelto nella fase precedente. Quindi scegli Esegui per eseguire la query.

```
MSCK REPAIR TABLE table_name;
```

7. Per cancellare l'editor di query, scegli Cancella. Esegui la seguente query `SELECT` per recuperare tutte le voci nel report di inventario originale e sostituire qualsiasi ID versione vuota con le stringhe `null`. Sostituisci il codice *`table_name`* con quello che hai scelto in precedenza

e sostituisci *YYYY-MM-DD-HH-MM* nella clausola WHERE con la data del report di inventario su cui eseguire questo strumento. Quindi scegli Esegui per eseguire la query.

```
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. Torna alla console di Amazon S3 (https://console.aws.amazon.com/s3/), e vai al bucket S3 che hai scelto in precedenza per la Posizione del risultato della query. All'interno, è presente una serie di cartelle che terminano con la data.

   Ad esempio, dovrebbe essere visualizzato un input simile a s3://***DOC-EXAMPLE-BUCKET***/***query-result-location***/Unsaved/2021/10/07/. Dovrebbe essere possibile visualizzare i file `.csv` contenenti i risultati della query SELECT che hai eseguito.

   Scegli il file CSV con la data di modifica più recente. Scarica questo file sul tuo computer locale per il passaggio successivo.

9. Il file CSV generato contiene una riga di intestazione. Per utilizzare questo file CSV come input per un processo S3 Batch Operations, è necessario rimuovere la riga di intestazione, poiché Batch Operations non supporta le righe di intestazione nei manifest CSV.

   Per rimuovere la riga di intestazione, è possibile eseguire uno dei seguenti comandi sul file. Sostituisci *file.csv* con il nome del tuo file CSV.

   Per macchine macOS e Linux, eseguire il comando `tail` in una finestra Terminal.

   ```
   tail -n +2 file.csv > tmp.csv && mv tmp.csv file.csv
   ```

   Per macchine Windows, eseguire il seguente script in una finestra di Windows PowerShell. Sostituire *File-location* con il percorso al file e *file.csv* con il nome del file.

   ```
   $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. Dopo aver rimosso la riga di intestazione dal file CSV, è possibile utilizzarla come manifest in un processo di S3 Batch Operations. Carica il file CSV in un bucket S3 o in una posizione a tua scelta, quindi crea un processo Batch Operations utilizzando il file CSV come manifest.

    Per ulteriori informazioni sulla creazione di un processo di Batch Operations, consulta Creazione di un processo di operazioni in batch S3.

Utilizzo del campo ACL oggetto

Un report Inventario Amazon S3 contiene un elenco degli oggetti presenti nel bucket di origine S3 e i metadata per ogni oggetto. Il campo Elenco di controllo di accesso dell'oggetto (ACL) è un campo di metadati disponibile in Inventario Amazon S3. In particolare, il campo ACL oggetto contiene la lista di controllo degli accessi (ACL) per ciascun oggetto. L'ACL di un oggetto definisce a quali Account AWS o gruppi è concesso l'accesso a questo oggetto e il tipo di accesso concesso. Per ulteriori informazioni, consulta Panoramica delle liste di controllo accessi (ACL) e Elenco di Amazon S3 Inventory.

Il campo ACL oggetto nei report di Inventario Amazon S3 è definito in formato JSON. I dati JSON includono i seguenti campi:

- `version`: la versione del formato del campo ACL oggetto nei report di inventario. È in formato data `yyyy-mm-dd`.
- `status`: i valori possibili sono `AVAILABLE` o `UNAVAILABLE` per indicare se un ACL oggetto ACL è disponibile per un oggetto. Quando lo stato dell'ACL oggetto è `UNAVAILABLE`, il valore del campo Proprietario dell'oggetto nel report di inventario è anche `UNAVAILABLE`.
- `grants`: coppie autorizzate dall'assegnatario che elencano lo stato di autorizzazione di ciascun assegnatario che viene concesso dall'ACL oggetto. I valori disponibili per un assegnatario sono `CanonicalUser` e `Group`. Per ulteriori informazioni sugli assegnatari, consulta Assegnatari nelle liste di controllo degli accessi.

Per un assegnatario con il tipo `Group`, una coppia autorizzata dall'assegnatario include i seguenti attributi:

- `uri`: un gruppo Amazon S3 predefinito.
- `permission`: le autorizzazioni ACL che vengono concesse sull'oggetto. Per ulteriori informazioni, consulta Autorizzazioni ACL su un oggetto.
- `type`: il tipo`Group`, che denota che l'assegnatario è un gruppo.

Per un assegnatario con il tipo `CanonicalUser`, una coppia autorizzata dall'assegnatario include i seguenti attributi:

- `canonicalId`: una forma offuscata dell'ID Account AWS . L'ID utente canonico di an Account AWS è specifico di quell'account. Puoi recuperare l'ID utente canonico. Per ulteriori informazioni, consulta Trova l'ID utente canonico per il tuo Account AWS nella Guida di riferimento per la gestione dell'AWS account.

  > **Note**
  >
  > Se un beneficiario in un ACL è l'indirizzo e-mail di un beneficiario Account AWS, S3 Inventory utilizza tale indirizzo Account AWS e il `canonicalId` `CanonicalUser` tipo per specificare questo beneficiario. Per ulteriori informazioni, consulta Assegnatari nelle liste di controllo degli accessi.

- `permission`: le autorizzazioni ACL che vengono concesse sull'oggetto. Per ulteriori informazioni, consulta Autorizzazioni ACL su un oggetto.
- `type`— Il tipo`CanonicalUser`, che indica che il beneficiario è un. Account AWS

L'esempio seguente mostra i possibili valori per il campo ACL oggetto in 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**
>
> Il campo ACL oggetto è definito in formato JSON. Un report di inventario visualizza il valore per il campo ACL oggetto come stringa con codifica base64.
> Ad esempio, si supponga di avere il seguente campo ACL oggetto in formato JSON:
>
> ```
> {
>         "version": "2022-11-10",
>         "status": "AVAILABLE",
>         "grants": [{
>             "canonicalId": "example-canonical-user-ID",
>             "type": "CanonicalUser",
>             "permission": "READ"
>         }]
> }
> ```
>
> Il campo ACL oggetto è codificato e visualizzato come la seguente stringa con codifica base64:
>
> ```
> eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3siY2Fub25pY2FsSW
> ```
>
> Per ottenere il valore decodificato in JSON per il campo ACL oggetto, puoi eseguire una query su questo campo in Amazon Athena. Per ulteriori esempi di query, consulta Esecuzione di query sull'inventario Amazon S3 con Amazon Athena.

Panoramica sulla replica degli oggetti

Puoi utilizzare la replica per abilitare la copia automatica e asincrona degli oggetti tra i bucket Amazon S3. I bucket configurati per la replica di oggetti possono essere di proprietà dello stesso Account AWS o di account diversi. Puoi replicare gli oggetti in un singolo bucket o in più bucket di destinazione. I bucket di destinazione possono trovarsi in una regione diversa Regioni AWS o all'interno della stessa regione del bucket di origine.

Esistono due tipi di replica: replica dal vivo e replica su richiesta.

- Replica live: per replicare automaticamente oggetti nuovi e aggiornati man mano che vengono scritti nel bucket di origine, utilizzate la replica live. La replica live non replica gli oggetti che esistevano nel bucket prima della configurazione della replica. Per replicare oggetti che esistevano prima di impostare la replica, utilizzate la replica su richiesta.
- Replica su richiesta: per replicare oggetti esistenti dal bucket di origine a uno o più bucket di destinazione su richiesta, utilizza S3 Batch Replication. Per ulteriori informazioni sulla replica di oggetti esistenti, consulta la sezione Quando utilizzare S3 Batch Replication.

Esistono due forme di replica in tempo reale: la replica tra regioni (CRR) e la replica a regione singola (SRR).

- Replica tra regioni (CRR): puoi utilizzare CRR per replicare oggetti su bucket Amazon S3 in diversi modi. Regioni AWS Per ulteriori informazioni sul CRR, consulta. the section called “Quando utilizzare la replica tra aree”
- Replica a regione singola (SRR): puoi utilizzare SRR per copiare oggetti tra bucket Amazon S3 nello stesso. Regione AWS Per ulteriori informazioni su SRR, consulta. the section called “Quando utilizzare la replica della stessa regione”

Argomenti

Perché utilizzare la replica?

La replica può essere utile per gli scopi seguenti:

- Replica di oggetti conservando i metadati: puoi utilizzare la replica per creare copie degli oggetti che conservano tutti i metadati, ad esempio l'ora di creazione dell'oggetto originale e gli ID della versione. Questa funzionalità è importante se vuoi assicurarti che la replica sia identica all'oggetto di origine.
- Replica di oggetti in classi di archiviazione diverse: puoi utilizzare la replica per inserire direttamente gli oggetti in S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive o in un'altra classe di archiviazione nei bucket di destinazione. Puoi anche replicare i dati nella stessa classe di archiviazione e utilizzare configurazioni del ciclo di vita nei bucket di destinazione per spostare gli oggetti in una classe di archiviazione più inattiva col passare del tempo.
- Conserva le copie degli oggetti con proprietà diverse: indipendentemente dal proprietario dell'oggetto di origine, puoi chiedere ad Amazon S3 di cambiare la proprietà della replica con Account AWS quella proprietaria del bucket di destinazione. Questa opzione è detta sostituzione del proprietario. Puoi utilizzare questa opzione per limitare l'accesso alle repliche degli oggetti.
- Conserva gli oggetti archiviati su più oggetti Regioni AWS: per garantire differenze geografiche nella posizione in cui vengono conservati i dati, puoi impostare più bucket di destinazione tra diversi. Regioni AWS Questa funzionalità potrebbe aiutarti a soddisfare determinati requisiti di conformità.
- Replica gli oggetti entro 15 minuti: per replicare i dati nella stessa regione Regione AWS o in regioni diverse entro un periodo di tempo prevedibile, puoi utilizzare S3 Replication Time Control (S3 RTC). S3 RTC replica il 99,99% dei nuovi oggetti archiviati in Amazon S3 entro 15 minuti, secondo un Accordo sul Livello di Servizio (SLA). Per ulteriori informazioni, consulta the section called “Utilizzo di S3 Replication Time Control”.

  > **Note**
  >
  > S3 RTC non si applica a Batch Replication. Batch Replication è un processo di replica on demand e può essere monitorato con S3 Batch Operations. Per ulteriori informazioni, consulta Monitoraggio dei rapporti sullo stato e sul completamento dei processi.

- Sincronizzazione di bucket, replica di oggetti esistenti e replica di oggetti falliti o replicati in precedenza: per sincronizzare i bucket e replicare oggetti esistenti, utilizza Batch Replication come operazione di replica on demand. Per ulteriori informazioni sul momento in cui è necessario utilizzare Batch Replication, consulta la sezione Quando utilizzare S3 Batch Replication.

- Replicare gli oggetti ed eseguire il failover su un bucket all'interno di un altro Regione AWS: per mantenere sincronizzati tutti i metadati e gli oggetti tra i bucket durante la replica dei dati, utilizza le regole di replica bidirezionale prima di configurare i controlli di failover del punto di accesso multi-regione Amazon S3. Le regole di replica bidirezionale aiutano a garantire che quando i dati vengono scritti nel bucket S3, il traffico viene poi replicato nuovamente nel bucket di origine.

Quando utilizzare la replica tra aree

La replica tra regioni (Cross-Region Replication, CRR) S3 viene utilizzata per copiare gli oggetti tra bucket Amazon S3 in Regioni AWS diverse. La replica CRR consente di completare le seguenti operazioni:

- Rispetto dei requisiti di conformità: sebbene di default Amazon S3 archivi i dati in più zone di disponibilità geograficamente distanti, per soddisfare i requisiti di conformità potrebbe essere necessario archiviarli a distanze ancora maggiori. Per soddisfare questi requisiti, puoi utilizzare la replica tra regioni e replicare i dati tra Regioni AWS distanti.
- Ridurre al minimo la latenza: se i clienti si trovano in due aree geografiche, è possibile ridurre al minimo la latenza nell'accesso agli oggetti conservando le copie degli oggetti in luoghi geograficamente più vicini agli utenti. Regioni AWS
- Aumenta l'efficienza operativa: se disponi di cluster di elaborazione in due aree diverse Regioni AWS che analizzano lo stesso set di oggetti, puoi scegliere di conservare le copie degli oggetti in tali regioni.

Quando utilizzare la replica della stessa regione

La replica nella stessa regione (Same-Region Replication, SRR) viene utilizzata per copiare gli oggetti tra bucket Amazon S3 nella stessa Regione AWS. La replica SRR consente di completare le seguenti operazioni:

- Aggregazione dei registri in un solo bucket: se archivi i registri in più bucket o in più account, puoi replicarli facilmente in un solo bucket nella tua regione. Questo semplifica l'elaborazione dei registri in una sola posizione.
- Configurazione della replica in tempo reale tra gli account di produzione e test: se tu o i tuoi clienti disponete di account di produzione e test che utilizzano gli stessi dati, potete replicare gli oggetti tra più account conservandone i metadati.

- Rispetta le leggi sulla sovranità dei dati: potrebbe essere necessario archiviare più copie dei dati in modo separato Account AWS all'interno di una determinata regione. La replica nella stessa regione può aiutarti a eseguire la replica automatica di dati fondamentali nel caso in cui i regolamenti di conformità non consentano ai dati di lasciare il tuo Paese.

Quando utilizzare la replica bidirezionale

- Crea set di dati condivisi su più oggetti Regioni AWS: con la sincronizzazione delle modifiche alle repliche, puoi replicare facilmente le modifiche ai metadati, come le liste di controllo degli accessi agli oggetti (ACL), i tag degli oggetti o i blocchi degli oggetti, sugli oggetti di replica. Questa replica bidirezionale è importante se si desidera mantenere sincronizzati tutti gli oggetti e le modifiche ai metadati degli oggetti. È possibile abilitare la sincronizzazione delle modifiche delle repliche su una regola di replica nuova o esistente quando si esegue una replica bidirezionale tra due o più bucket nella stessa o in diverse Regioni AWS.
- Mantieni i dati sincronizzati tra le regioni durante il failover: puoi sincronizzare i dati in bucket tra loro configurando regole di replica bidirezionale con S3 Cross-Region Replication (CRR) direttamente Regioni AWS da un punto di accesso multiregionale. Per prendere una decisione informata su quando avviare il failover, puoi anche abilitare i parametri di replica S3 in modo da monitorare la replica in Amazon CloudWatch, in S3 Replication Time Control (S3 RTC) o dal punto di accesso multiregionale.
- Rendere la tua applicazione altamente disponibile: anche in caso di interruzione del traffico regionale, puoi utilizzare regole di replica bidirezionale per mantenere sincronizzati tutti i metadati e gli oggetti tra i bucket durante la replica dei dati.

Quando utilizzare S3 Batch Replication

Batch Replication replica gli oggetti esistenti in bucket diversi come opzione on demand. A differenza della replica in tempo reale, questi processi possono essere eseguiti all'occorrenza. Batch Replication può essere utile per gli scopi seguenti:

- Replica di oggetti esistenti: è possibile utilizzare Batch Replication per replicare gli oggetti aggiunti al bucket prima della configurazione della replica nella stessa regione o della replica tra regioni.
- Replica di oggetti che in precedenza non sono stati replicati: è possibile applicare un filtro a un processo Batch Replication per tentare di replicare gli oggetti con uno stato di replica FAILED (Fallito).

- Replica di oggetti già replicati: potrebbe essere necessario archiviare più copie dei dati in Account AWS o Regioni AWS separati. Batch Replication può replicare gli oggetti esistenti nelle destinazioni appena aggiunte.
- Replica di repliche di oggetti creati da una regola di replica: le configurazioni di replica creano repliche di oggetti nei bucket di destinazione. Le repliche di oggetti possono essere replicate solo con Batch Replication.

Requisiti del carico di lavoro e replica in tempo reale

A seconda dei requisiti del carico di lavoro, alcuni tipi di replica saranno più adatti al vostro caso d'uso rispetto ad altri. Utilizza la tabella seguente per determinare il tipo di replica da utilizzare per la tua situazione e se utilizzare S3 Replication Time Control (S3 RTC) per il tuo carico di lavoro. S3 RTC replica il 99,99 percento dei nuovi oggetti archiviati in Amazon S3 entro 15 minuti (supportato da un accordo sul livello di servizio o SLA). Per ulteriori informazioni, consulta the section called “Utilizzo di S3 Replication Time Control”.

Requisiti del carico di lavoro per il confronto delle repliche

Requisiti del carico di lavoro	S3 RTC (SLA di 15 minuti)	Replica tra regioni (CRR)	Replica in un'unica regione (SRR)
Replica oggetti tra diversi Account AWS	Sì	Sì	Sì
Replica gli oggetti all'interno degli stessi Regione AWS entro 24-48 ore (senza supporto SLA)	No	No	Sì
Replica oggetti tra diversi Regioni AWS entro 24-48 ore (senza supporto SLA)	No	Sì	No
Tempo di replica prevedibile: supportat o dallo SLA per	Sì	No	No

Requisiti del carico di lavoro	S3 RTC (SLA di 15 minuti)	Replica tra regioni (CRR)	Replica in un'unica regione (SRR)
replicare il 99,9% degli oggetti entro 15 minuti			

Cosa replica Amazon S3?

Amazon S3 replica solo elementi specifici nei bucket configurati per la replica.

Argomenti

Che cosa viene replicato con le configurazioni di replica?

Di default, Amazon S3 replica quanto segue:

- Oggetti creati dopo l'aggiunta di una configurazione di replica.
- Oggetti non crittografati.
- Oggetti crittografati utilizzando chiavi fornite dal cliente (SSE-C), oggetti crittografati a riposo con una chiave gestita Amazon S3 (SSE-S3) o una chiave KMS archiviata in (SSE-KMS). AWS Key Management Service Per ulteriori informazioni, consulta the section called “Replica di oggetti crittografati”.
- Metadati dell'oggetto dagli oggetti di origine alle repliche. Per informazioni sulla replica dei metadati dalle repliche agli oggetti di origine, consulta Replica delle modifiche ai metadati con la sincronizzazione delle modifiche alla replica Amazon S3.
- Solo gli oggetti nel bucket di origine per cui il proprietario del bucket dispone delle autorizzazioni di lettura degli oggetti e delle liste di controllo accessi (ACL).

  Per ulteriori informazioni sulla proprietà delle risorse, consulta Proprietà di bucket e oggetti di Amazon S3.

- Gli aggiornamenti delle liste di controllo accessi degli oggetti, a meno che non indichi ad Amazon S3 di modificare il proprietario della replica quando i bucket di origine e di destinazione non sono di proprietà degli stessi account.

  Per ulteriori informazioni, consulta Modifica del proprietario della replica.

  Per la sincronizzazione delle due liste di controllo accessi da parte di Amazon S3 potrebbe essere necessario del tempo. Questa modifica di proprietà si applica solo agli oggetti creati dopo che è stata aggiunta una configurazione di replica al bucket.
- Eventuali tag degli oggetti.
- Eventuali informazioni sulla conservazione del blocco oggetti S3.

  Quando Amazon S3 replica gli oggetti con informazioni sulla conservazione, applica gli stessi controlli di conservazione alle repliche, ignorando il periodo di conservazione predefinito configurato sui bucket di destinazione. Se non sono previsti controlli di conservazione applicati agli oggetti nel bucket di origine e la replica viene effettuata nei bucket di destinazione con un periodo di conservazione predefinito impostato, il periodo di conservazione predefinito dei bucket di destinazione viene applicato alle repliche degli oggetti. Per ulteriori informazioni, consulta Utilizzo del blocco oggetti S3.

Effetto delle operazioni di eliminazione sulla replica

Se si elimina un oggetto dal bucket di origine, per impostazione predefinita si verificano le seguenti azioni:

- Se effettui una richiesta di eliminazione (DELETE) senza specificare l'ID della versione dell'oggetto, Amazon S3 aggiunge un contrassegno di eliminazione. Amazon S3 gestisce il contrassegno di eliminazione in questo modo:
  - Se usi la versione più recente della configurazione di replica (ovvero, se specifichi l'elemento `Filter` in una regola di configurazione di replica), Amazon S3 non replica automaticamente il contrassegno di eliminazione. Tuttavia, puoi aggiungere la replica dei marker di eliminazione alle regole. non-tag-based Per ulteriori informazioni, consulta Replica dei contrassegni di eliminazione tra i bucket.
  - Se non si specifica l'elemento `Filter`, Amazon S3 presuppone che la configurazione di replica sia la versione V1 e replica i contrassegni di eliminazione derivanti dalle azioni dell'utente. Tuttavia, se Amazon S3 elimina un oggetto a causa di un'azione del ciclo di vita, il contrassegno di eliminazione non viene replicato nei bucket di destinazione.

- Se nella richiesta `DELETE` specifichi l'ID della versione dell'oggetto da eliminare, Amazon S3 elimina la versione dell'oggetto nel bucket di origine. ma non replica l'eliminazione nei bucket di destinazione. In altre parole, non elimina la stessa versione dell'oggetto dai bucket di destinazione. Ciò permette di proteggere i dati da eliminazioni da parte di utenti malintenzionati.

Che cosa non viene replicato con le configurazioni di replica?

Di default, Amazon S3 non replica quanto segue:

- Gli oggetti nel bucket di origine che sono repliche create da un'altra regola di replica. Supponiamo, per esempio, di configurare una replica dove il bucket A è l'origine e il bucket B è la destinazione. Supponiamo ora di aggiungere un'altra configurazione di replica dove il bucket B è l'origine e il bucket C è la destinazione. In questo caso, gli oggetti nel bucket B che sono repliche di oggetti nel bucket A non vengono replicati nel bucket C.

  Per replicare oggetti che sono repliche, utilizza Batch Replication. Per ulteriori informazioni sulla configurazione di Batch Replication, visita Replica di oggetti esistenti.
- Oggetti nel bucket di origine che sono già stati replicati in una destinazione diversa. Se, ad esempio, modifichi il bucket di destinazione in una configurazione di replica esistente, Amazon S3 non replica di nuovo gli oggetti.

  Per replicare oggetti replicati in precedenza, utilizza Batch Replication. Per ulteriori informazioni sulla configurazione di Batch Replication, visita Replica di oggetti esistenti.
- La replica batch non supporta la ripetizione della replica di oggetti eliminati con l'ID versione dell'oggetto dal bucket di destinazione. Per replicare nuovamente questi oggetti è possibile copiare gli oggetti di origine presenti con un processo di copia in batch. La copia di tali oggetti crea nuove versioni dell'oggetto nel bucket di origine e avvia automaticamente la replica nella destinazione. Per ulteriori informazioni su come utilizzare la copia batch, consulta Esempi che utilizzano operazioni in batch per copiare oggetti.
- Per impostazione predefinita, quando si esegue la replica da un altro Account AWS, i marker di eliminazione aggiunti al bucket di origine non vengono replicati.

  Per informazioni su come replicare i contrassegni di eliminazione, consulta la sezione Replica dei contrassegni di eliminazione tra i bucket.
- Oggetti archiviati nelle classi o nei livelli di storage S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, S3 Intelligent-Tiering Archive Access o S3 Intelligent-Tiering Deep Archive Access.

Non è possibile replicare questi oggetti finché non vengono ripristinati e copiati in una classe di archiviazione diversa.

Per ulteriori informazioni su S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive, consulta. Classi di archiviazione per oggetti a cui si accede raramente

Per saperne di più su S3 Intelligent-Tiering, consulta. Amazon S3 Intelligent-Tiering

- Oggetti nel bucket di origine per cui il proprietario del bucket non dispone di autorizzazioni sufficienti per eseguire la replica.

  Per informazioni su come il proprietario di un oggetto può concedere le autorizzazioni al proprietario del bucket, consulta la sezione Concedere autorizzazioni multi-account per il caricamento di oggetti a garanzia del controllo completo da parte del proprietario del bucket.

- Aggiornamenti alle risorse secondarie a livello di bucket.

  Se, ad esempio, modifichi la configurazione del ciclo di vita o aggiungi una configurazione di notifica nel bucket di origine, tali modifiche non vengono applicate nel bucket di destinazione. Questa funzionalità permette la presenza di configurazioni diverse nei bucket di origine e di destinazione.

- Operazioni eseguite dalla configurazione del ciclo di vita.

  Ad esempio, se la configurazione del ciclo di vita è abilitata solo nel bucket di origine, Amazon S3 crea i contrassegni di eliminazione per gli oggetti scaduti, ma non replica i contrassegni. Per applicare al bucket di origine e a quello di destinazione la stessa configurazione del ciclo di vita, è sufficiente abilitare quest'ultima in entrambi. Per ulteriori informazioni sulla configurazione del ciclo di vita, consulta Gestione del ciclo di vita dello storage.

- Quando si utilizzano regole di replica basate su tag con replica live, i nuovi oggetti devono essere etichettati con il tag della regola di replica corrispondente durante l'operazione. `PutObject` Altrimenti, gli oggetti non verranno replicati. Se gli oggetti vengono etichettati dopo l'`PutObject`operazione, anche tali oggetti non verranno replicati.

  Per replicare oggetti che sono stati etichettati dopo l'`PutObject`operazione, è necessario utilizzare S3 Batch Replication. Per ulteriori informazioni su Batch Replication, consulta la sezione Replica di oggetti esistenti.

In che modo la crittografia predefinita del bucket influisce sulla replica

Una volta abilitata la crittografia predefinita per un bucket di destinazione della replica, si applica il seguente comportamento di crittografia:

- Se gli oggetti nel bucket di origine non sono crittografati, gli oggetti replicati nel bucket di destinazione vengono crittografati in base alle impostazioni di crittografia predefinita del bucket di destinazione. Di conseguenza, i tag di entità (ETag) degli oggetti di origine differiscono dagli ETag degli oggetti di replica. Se disponi di applicazioni che utilizzano ETag, devi aggiornarle per tenere conto di questa differenza.
- Se gli oggetti nel bucket di origine sono crittografati utilizzando la crittografia lato server con chiavi gestite Amazon S3 (SSE-S3), la crittografia lato server con chiavi () (SSE-KMS AWS KMS) o la crittografia lato server a doppio livello con AWS Key Management Service AWS KMS chiavi (DSSE-KMS), gli oggetti di replica nel bucket di destinazione utilizzano lo stesso tipo di crittografia degli oggetti di origine. Le impostazioni della crittografia predefinita del bucket di destinazione non vengono utilizzate.

Requisiti e considerazioni per la replica

La replica di Amazon S3 richiede quanto segue:

- Il proprietario del bucket di origine deve avere l'origine e la destinazione Regioni AWS abilitate per il proprio account. La regione di destinazione deve essere abilitata per l'account del proprietario del bucket.

  Per ulteriori informazioni sull'attivazione o la disabilitazione di un Regione AWS, consulta Gestione Regioni AWS in. Riferimenti generali di AWS
- Sia per il bucket di origine che per quello di destinazione deve essere abilitata la funzione Controllo delle versioni. Per ulteriori informazioni sulla funzione Controllo delle versioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.
- Amazon S3 deve disporre delle autorizzazioni necessarie per replicare gli oggetti dal bucket di origine a quelli di destinazione per tuo conto. Per ulteriori informazioni su queste autorizzazioni, consulta la sezione Impostazione delle autorizzazioni per la replica in tempo reale.
- Se il proprietario del bucket di origine non possiede l'oggetto nel bucket, il proprietario dell'oggetto deve concedere al proprietario del bucket le autorizzazioni `READ` e `READ_ACP` con la lista di controllo degli accessi (ACL) dell'oggetto. Per ulteriori informazioni, consulta Panoramica delle liste di controllo accessi (ACL).

- Se il blocco oggetti S3 è abilitato nel bucket di origine, deve essere abilitato anche nei bucket di destinazione.

  Per abilitare la replica su un bucket con Object Lock abilitato, devi utilizzare l' AWS Command Line Interface API REST o gli SDK. AWS Per ulteriori informazioni generali, consulta Utilizzo del blocco oggetti S3.

  > **Note**
  >
  > È necessario concedere due nuove autorizzazioni sul bucket S3 di origine nel ruolo AWS Identity and Access Management (IAM) utilizzato per configurare la replica. Le due nuove autorizzazioni sono `s3:GetObjectRetention` e `s3:GetObjectLegalHold`. Se il ruolo dispone di un'autorizzazione `s3:Get*`, soddisfa il requisito. Per ulteriori informazioni, consulta Impostazione delle autorizzazioni per la replica in tempo reale.

Per ulteriori informazioni, consultare Configurazione della replica in tempo reale.

Quando imposti la configurazione di replica in uno scenario con più account, in cui il bucket di origine e quello di destinazione sono di proprietà di Account AWS diversi, si applica il seguente requisito aggiuntivo:

- Il proprietario dei bucket di destinazione deve concedere al proprietario del bucket di origine le autorizzazioni necessarie per replicare gli oggetti con una policy del bucket. Per ulteriori informazioni, consulta Concessione delle autorizzazioni quando i bucket di origine e di destinazione sono di proprietà di diversi Account AWS.
- I bucket di destinazione non possono essere configurati come bucket con pagamento a carico del richiedente. Per ulteriori informazioni, consulta Utilizzo dei bucket con pagamento a carico del richiedente per utilizzo e trasferimenti di storage.

Considerazioni sulla replica

Prima di creare una configurazione di replica, tenete presente le seguenti considerazioni.

Argomenti

- Configurazione del ciclo di vita e repliche di oggetti
- Configurazione della funzione Controllo delle versioni e configurazione di replica

- Utilizzo di Replica S3 con Piano intelligente Amazon S3
- Configurazione della registrazione e configurazione di replica
- CRR e regione di destinazione
- Replica in batch S3
- Controllo del tempo di replica S3

Configurazione del ciclo di vita e repliche di oggetti

Il tempo richiesto da Amazon S3 per la replica di un oggetto dipende dalle dimensioni dell'oggetto. Per gli oggetti di grandi dimensioni, questa operazione può richiedere anche diverse ore. Anche se la replica può richiedere tempo prima di diventare disponibile nella destinazione, il tempo necessario per creare la replica corrisponde a quello che è stato necessario per creare l'oggetto corrispondente nel bucket di origine. Se una configurazione del ciclo di vita è abilitata in un bucket di destinazione, le regole del ciclo di vita rispettano l'ora di creazione originale dell'oggetto, non l'ora in cui la replica è diventata disponibile nel bucket di destinazione.

La configurazione di replica richiede che nel bucket sia abilitata la funzione Controllo delle versioni. Quando si abilita tale funzione in un bucket, tenere presente quanto segue:

- Se è presente una configurazione del ciclo di vita di scadenza dell'oggetto, dopo avere abilitato la funzione Controllo delle versioni, è necessario aggiungere una policy `NonCurrentVersionExpiration` per mantenere lo stesso comportamento di eliminazione permanente presente prima dell'abilitazione della funzione.
- Se è presente una configurazione del ciclo di vita di transizione, dopo avere abilitato la funzione Controllo delle versioni, è consigliabile aggiungere una policy `NonCurrentVersionTransition`.

Configurazione della funzione Controllo delle versioni e configurazione di replica

Quando si configura la replica in un bucket, la funzione Controllo delle versioni deve essere abilitata sia nel bucket di origine che in quello di destinazione. Dopo avere abilitato la funzione Controllo delle versioni in entrambi i bucket di origine e di destinazione e avere configurato la replica nel bucket di origine, potrebbero verificarsi i problemi seguenti:

- Se si tenta di disabilitare la funzione Controllo delle versioni nel bucket di origine, Amazon S3 restituisce un errore. Prima di poter disabilitare la funzione Controllo delle versioni nel bucket di origine, è necessario rimuovere la configurazione di replica.

- Se si disabilita la funzione Controllo delle versioni nel bucket di destinazione, la replica ha esito negativo. Lo stato della replica dell'oggetto di origine è `FAILED`.

Utilizzo di Replica S3 con Piano intelligente Amazon S3

Piano intelligente Amazon S3 è una classe di storage progettata per ottimizzare i costi di archiviazione spostando automaticamente i dati nel livello di accesso più conveniente. Per un monitoraggio degli oggetti mensile e una tariffa di automazione bassi, S3 Intelligent-Tiering monitora i modelli di accesso e sposta automaticamente gli oggetti ai quali non è stato eseguito l'accesso a livelli di accesso a costo più basso.

La replica di oggetti archiviati in Piano intelligente Amazon S3 con S3 Batch Replication o richiamando CopyObject o UploadPartCopycostituisce accesso. In questi casi, gli oggetti di origine delle operazioni di copia o replica sono suddivisi su più livelli.

Per ulteriori informazioni sul Piano intelligente Amazon S3, consulta Amazon S3 Intelligent-Tiering.

Configurazione della registrazione e configurazione di replica

Se Amazon S3 invia log a un bucket in cui è abilitata la replica, gli oggetti dei log vengono replicati.

Se nel bucket di origine o di destinazione sono abilitati i log di accesso al server (Registrazione delle richieste con registrazione dell'accesso al server) o i log AWS CloudTrail (Registrazione delle chiamate API Amazon S3 tramite AWS CloudTrail), Amazon S3 include nei log le richieste correlate alla replica. Ad esempio, Amazon S3 include nei log ogni oggetto che replica.

CRR e regione di destinazione

Amazon S3 Cross-Region Replication (CRR) viene utilizzato per copiare oggetti tra bucket S3 in diversi. Regioni AWS Puoi scegliere la regione del bucket di destinazione in base alle esigenze aziendali o a considerazioni sui costi. Ad esempio, i costi di trasferimento dei dati tra regioni variano in base alle regioni scelte.

Supponiamo che scegli Stati Uniti orientali (Virginia settentrionale) (us-east-1) come regione per il bucket di origine. Se scegli Stati Uniti occidentali (Oregon) (us-west-2) come regione per i bucket di destinazione, il costo sarà maggiore di quanto sarebbe scegliendo la regione Stati Uniti orientali (Ohio) (us-east-2). Per informazioni sui prezzi, consulta la sezione relativa ai prezzi per il trasferimento dati in Prezzi di Amazon S3.

La replica nella stessa regione non prevede costi per il trasferimento dei dati

Replica in batch S3

Per informazioni sulle considerazioni relative alla replica in batch, vedere. Considerazioni su S3 Batch Replication

Controllo del tempo di replica S3

Per informazioni sulle best practice e sulle considerazioni relative a S3 Replication Time Control (S3 RTC), consulta. Best practice e linee guida per S3 RTC

Configurazione della replica in tempo reale

> **Note**
>
> Gli oggetti esistenti prima della configurazione della replica non vengono replicati automaticamente. In altre parole, Amazon S3 non esegue la replica retroattiva di oggetti. Per replicare oggetti creati prima della configurazione della replica, utilizza S3 Batch Replication. Per ulteriori informazioni sulla configurazione di Batch Replication, visita Replica di oggetti esistenti.

Per abilitare la replica in tempo reale, Same-Region Replication (SRR) o Cross-Region Replication (CRR), aggiungi una configurazione di replica al tuo bucket di origine. Questa configurazione indica ad Amazon S3 di replicare gli oggetti come specificato. Nella configurazione di replica, è necessario fornire le informazioni seguenti:

- I bucket di destinazione: uno o più bucket in cui desideri che Amazon S3 replichi gli oggetti.
- Gli oggetti da replicare: puoi replicare tutti gli oggetti presenti nel bucket di origine o solo una parte di essi. Puoi identificare un sottoinsieme specificando nella configurazione un prefisso di nome di chiave, uno o più tag di oggetti oppure entrambi.

  Se, ad esempio, configuri una regola di replica per replicare solo gli oggetti con il prefisso di nome di chiave `Tax/`, Amazon S3 replica gli oggetti con chiavi come `Tax/doc1` o `Tax/doc2`. Ma non replica un oggetto con la chiave `Legal/doc3`. Se specifichi sia un prefisso sia uno o più tag, Amazon S3 replica solo gli oggetti con il prefisso della chiave e i tag specificati.
- Un ruolo AWS Identity and Access Management (IAM): Amazon S3 assume questo ruolo IAM per replicare gli oggetti per tuo conto.

Oltre a questi requisiti minimi, puoi scegliere tra le opzioni seguenti:

- Classe di archiviazione della replica: di default, Amazon S3 archivia le repliche di oggetti utilizzando la stessa classe di archiviazione dell'oggetto di origine. È possibile specificare una classe di storage diversa per le repliche.
- Proprietà della replica: Amazon S3 presuppone che la replica di un oggetto continuerà ad appartenere al proprietario dell'oggetto di origine. Quindi, quando replica gli oggetti, ne replica anche la lista di controllo degli accessi (ACL) corrispondente o l'impostazione S3 Object Ownership. Se i bucket di origine e di destinazione sono di proprietà di Account AWS diversi, è possibile configurare la replica in modo da assegnare la proprietà di una replica all' Account AWS proprietario del bucket di destinazione.

Puoi configurare la replica utilizzando l'API REST, AWS gli SDK AWS Command Line Interface (AWS CLI) o la console Amazon S3.

Amazon S3 fornisce anche delle operazioni API per supportare la configurazione delle regole di replica. Per ulteriori informazioni, consulta i seguenti argomenti nella Documentazione di riferimento delle API di Amazon Simple Storage Service:

- PutBucketReplication
- GetBucketReplication
- DeleteBucketReplication

Argomenti

- Configurazione di replica
- Impostazione delle autorizzazioni per la replica in tempo reale
- Esempi di configurazione della replica in tempo reale

Configurazione di replica

Amazon S3 archivia la configurazione di replica come file XML. Nel file XML di configurazione della replica, si specifica un ruolo AWS Identity and Access Management (IAM) e una o più regole.

```
<ReplicationConfiguration>
    <Role>IAM-role-ARN</Role>
    <Rule>
        ...
    </Rule>
```

```
    <Rule>
         ...
    </Rule>
     ...
</ReplicationConfiguration>
```

Amazon S3 non può replicare oggetti senza la tua autorizzazione. Le autorizzazioni vengono concesse con il ruolo IAM specificato nella configurazione di replica. Amazon S3 assume il ruolo IAM per replicare gli oggetti per tuo conto. È necessario innanzitutto concedere le autorizzazioni necessarie al ruolo IAM. Per ulteriori informazioni sulla gestione delle autorizzazioni, consulta la sezione Impostazione delle autorizzazioni per la replica in tempo reale.

Puoi aggiungere una regola a una configurazione di replica quando:

- Vuoi replicare tutti gli oggetti.
- Vuoi replicare un sottoinsieme di oggetti. Identifichi il sottoinsieme di oggetti aggiungendo un filtro alla regola. Nel filtro specifichi un prefisso di chiave o tag dell'oggetto o una combinazione di questi elementi, per identificare il sottoinsieme di oggetti a cui si applica la regola. I filtri si applicano agli oggetti che corrispondono ai valori esatti specificati.

Per replicare più sottoinsiemi di oggetti, aggiungi diverse regole a una configurazione di replica. In ogni regola puoi specificare un filtro tramite cui selezionare un particolare sottoinsieme di oggetti. Puoi ad esempio scegliere di replicare gli oggetti con prefissi della chiave `tax/` o `document/`. Per fare ciò devi aggiungere due regole: una che specifica il filtro prefisso della chiave `tax/` e un'altra che specifica il prefisso della chiave `document/`. Per ulteriori informazioni sui prefissi della chiave dell'oggetto, consulta Organizzazione degli oggetti utilizzando i prefissi.

Nelle sezioni seguenti vengono fornite informazioni aggiuntive.

Argomenti

Configurazione di base delle regole

Ogni regola deve includere lo stato e la priorità della stessa. La regola deve anche indicare se replicare i contrassegni di eliminazione.

- `Status` indica se la regola è abilitata o disabilitata utilizzando i valori `Enabled` o `Disabled`. Se una regola è disabilitata, Amazon S3 non esegue le operazioni in essa specificate.
- `Priority` indica quale regola ha la precedenza ogni volta che due o più regole di replica sono in conflitto. Amazon S3 prova a replicare gli oggetti in base a tutte le regole di replica. Tuttavia, se esistono due o più regole con lo stesso bucket di destinazione, gli oggetti vengono replicati in base alla regola con la priorità più alta. Più elevato è il numero, maggiore è la priorità.
- `DeleteMarkerReplication` indica se replicare i contrassegni di eliminazione tramite i valori `Enabled` o `Disabled`.

Nella configurazione di destinazione devi specificare il nome del bucket in cui Amazon S3 deve replicare gli oggetti.

Nell'esempio seguente sono indicati i requisiti minimi per una regola V2. Per la compatibilità con le versioni precedenti, Amazon S3 continua a supportare il formato XML V1. Per ulteriori informazioni, consulta Compatibilità con le versioni precedenti.

```
...
    <Rule>
        <ID>Rule-1</ID>
        <Status>Enabled-or-Disabled</Status>
        <Filter>
            <Prefix></Prefix>
        </Filter>
        <Priority>integer</Priority>
        <DeleteMarkerReplication>
           <Status>Enabled-or-Disabled</Status>
        </DeleteMarkerReplication>
        <Destination>
           <Bucket>arn:aws:s3:::example-s3-bucket</Bucket>
        </Destination>
    </Rule>
    <Rule>
         ...
    </Rule>
     ...
...
```

Puoi anche specificare altre opzioni di configurazione. Ad esempio, puoi scegliere di utilizzare una classe di storage per le repliche degli oggetti diversa dalla classe dell'oggetto di origine.

Facoltativo: specifica di un filtro

Per scegliere un sottoinsieme di oggetti a cui si applica la regola, aggiungi un filtro facoltativo. Puoi filtrare in base al prefisso della chiave dell'oggetto, ai tag dell'oggetto o a una combinazione di entrambi. Se applichi un filtro in base sia al prefisso della chiave sia ai tag dell'oggetto, Amazon S3 combina i filtri utilizzando un operatore logico AND. In altre parole, la regola si applica a un sottoinsieme di oggetti con uno specifico prefisso della chiave e tag specifici.

Filtro in base al prefisso della chiave oggetto

Per specificare una regola con un filtro basato su un prefisso della chiave di un oggetto, utilizza il codice seguente. Puoi specificare un solo prefisso.

```
<Rule>
    ...
    <Filter>
        <Prefix>key-prefix</Prefix>
    </Filter>
    ...
</Rule>
...
```

Filtro basato su tag oggetto

Per specificare una regola con un filtro basato sui tag di un oggetto, utilizza il codice seguente. Puoi specificare uno o più tag dell'oggetto.

```
<Rule>
    ...
    <Filter>
        <And>
            <Tag>
                <Key>key1</Key>
                <Value>value1</Value>
            </Tag>
            <Tag>
                <Key>key2</Key>
                <Value>value2</Value>
```

```
            </Tag>
             ...
        </And>
    </Filter>
    ...
</Rule>
...
```

Filtro con un prefisso chiave e tag oggetto

Per specificare un filtro della regola con una combinazione di prefisso della chiave e tag di un oggetto, utilizza il codice seguente. I filtri vengono uniti in un elemento padre <And>. Amazon S3 esegue un'operazione logica AND per combinare questi filtri. In altre parole, la regola si applica a un sottoinsieme di oggetti con uno specifico prefisso della chiave e tag specifici.

```
<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**
>
> - Se specifichi una regola con un <Filter> elemento vuoto, la regola si applica a tutti gli oggetti nel bucket.
> - Quando si utilizzano regole di replica basate su tag con replica live, i nuovi oggetti devono essere etichettati con il tag della regola di replica corrispondente durante l'operazione.

`PutObject` Altrimenti, gli oggetti non verranno replicati. Se gli oggetti vengono etichettati dopo l'`PutObject`operazione, anche tali oggetti non verranno replicati.

Per replicare oggetti che sono stati etichettati dopo l'`PutObject`operazione, è necessario utilizzare S3 Batch Replication. Per ulteriori informazioni su Batch Replication, consulta la sezione Replica di oggetti esistenti.

Configurazioni di destinazione aggiuntive

Nella configurazione di destinazione devi specificare il bucket in cui Amazon S3 deve replicare gli oggetti. Puoi configurare la replica per replicare gli oggetti da un bucket di origine a uno solo o a più bucket di destinazione.

```
...
<Destination>
    <Bucket>arn:aws:s3:::example-s3-bucket</Bucket>
</Destination>
...
```

Puoi aggiungere le seguenti opzioni nell'elemento `<Destination>`.

Argomenti

- Specifica della classe di storage
- Aggiunta di più bucket di destinazione
- Specifica di parametri diversi per ogni regola di replica con più bucket di destinazione
- Modifica della proprietà della replica
- Abilitazione di S3 Replication Time Control
- Replica gli oggetti creati con la crittografia lato server utilizzando AWS KMS

Specifica della classe di storage

È possibile specificare la classe di storage per le repliche degli oggetti. Per impostazione predefinita, Amazon S3 utilizza la classe di storage dell'oggetto di origine per creare le repliche degli oggetti, come nell'esempio seguente.

```
...
<Destination>
```

```
        <Bucket>arn:aws:s3:::example-s3-bucket</Bucket>
        <StorageClass>storage-class</StorageClass>
</Destination>
...
```

Aggiunta di più bucket di destinazione

Puoi aggiungere più bucket di destinazione in una singola configurazione di replica, come indicato di seguito.

```
...
<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>
...
```

Specifica di parametri diversi per ogni regola di replica con più bucket di destinazione

Quando si aggiungono più bucket di destinazione in una singola configurazione di replica, puoi specificare parametri diversi per ogni regola di replica, come indicato di seguito.

```
...
<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>
...
```

Modifica della proprietà della replica

Quando i bucket di origine e di destinazione non sono di proprietà degli stessi account, è possibile modificare la proprietà della replica con Account AWS quella proprietaria del bucket di destinazione. A questo scopo, aggiungi l'elemento `AccessControlTranslation`. Questo elemento assume il valore `Destination`.

```
...
<Destination>
   <Bucket>arn:aws:s3:::example-s3-bucket</Bucket>
   <Account>destination-bucket-owner-account-id</Account>
   <AccessControlTranslation>
       <Owner>Destination</Owner>
   </AccessControlTranslation>
</Destination>
...
```

Se non si aggiunge l'`AccessControlTranslation`elemento alla configurazione di replica, le repliche sono di proprietà dello stesso Account AWS proprietario dell'oggetto di origine. Per ulteriori informazioni, consulta Modifica del proprietario della replica.

Abilitazione di S3 Replication Time Control

Puoi abilitare S3 Replication Time Control (S3 RTC) nella configurazione di replica. S3 RTC replica la maggior parte degli oggetti in pochi secondi e il 99,99% degli oggetti entro 15 minuti, secondo un Accordo sul Livello di Servizio (SLA).

> **Note**
>
> Per `EventThreshold` e `Time` è accettato solo un valore valido di `<Minutes>15</Minutes>`.

```
...
<Destination>
  <Bucket>arn:aws:s3:::example-s3-bucket</Bucket>
  <Metrics>
    <Status>Enabled</Status>
    <EventThreshold>
      <Minutes>15</Minutes>
    </EventThreshold>
```

```
  </Metrics>
  <ReplicationTime>
    <Status>Enabled</Status>
    <Time>
      <Minutes>15</Minutes>
    </Time>
  </ReplicationTime>
</Destination>
...
```

Per ulteriori informazioni, consulta Rispetto dei requisiti di conformità utilizzando S3 Replication Time Control (S3 RTC). Per esempi di API, consulta PutBucketReplicationAmazon Simple Storage Service API Reference.

Replica gli oggetti creati con la crittografia lato server utilizzando AWS KMS

Il bucket di origine potrebbe contenere oggetti creati con la crittografia lato server utilizzando le chiavi AWS Key Management Service () (AWS KMS SSE-KMS). Per impostazione predefinita, Amazon S3 non replica questi oggetti. Puoi facoltativamente indicare ad Amazon S3 di replicare questi oggetti. Per farlo, innanzitutto abilita esplicitamente questa funzionalità aggiungendo l'elemento `SourceSelectionCriteria`. Quindi fornisci AWS KMS key (per il bucket Regione AWS di destinazione) da utilizzare per crittografare le repliche degli oggetti. I seguenti esempi mostrano come specificare questi elementi.

```
...
<SourceSelectionCriteria>
  <SseKmsEncryptedObjects>
    <Status>Enabled</Status>
  </SseKmsEncryptedObjects>
</SourceSelectionCriteria>
<Destination>
  <Bucket>arn:aws:s3:::example-s3-bucket</Bucket>
  <EncryptionConfiguration>
    <ReplicaKmsKeyID>AWS KMS key ID to use for encrypting object replicas</
ReplicaKmsKeyID>
  </EncryptionConfiguration>
</Destination>
...
```

Per ulteriori informazioni, consulta Replica di oggetti crittografati (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

Esempi di configurazioni di replica

Per iniziare, puoi aggiungere le configurazioni di replica di esempio seguenti al bucket, in base alle esigenze.

> **Important**
>
> Per aggiungere una configurazione di replica a un bucket, devi disporre dell'autorizzazione `iam:PassRole`. Questa autorizzazione permette di passare il ruolo IAM che concede le autorizzazioni di replica ad Amazon S3. Puoi specificare il ruolo IAM fornendo l'Amazon Resource Name (ARN) utilizzato nell'elemento `Role` nel file XML della configurazione di replica. Per ulteriori informazioni, consulta Concessione di autorizzazioni utente per il passaggio di un ruolo a Servizio AWS nella Guida per l'utente di IAM.

Example 1: Configurazione di replica con una regola

La configurazione di replica di base seguente specifica una regola. La regola specifica un ruolo IAM che Amazon S3 può assumere e un singolo bucket di destinazione per le repliche degli oggetti. Un valore di `Status` pari a `Enabled` indica che la regola è in vigore.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>

    <Destination><Bucket>arn:aws:s3:::example-s3-bucket</Bucket></Destination>

  </Rule>
</ReplicationConfiguration>
```

Per scegliere un sottoinsieme di oggetti da replicare, puoi aggiungere un filtro. Nella configurazione seguente il filtro specifica un prefisso della chiave di un oggetto. Questa regola si applica agli oggetti con il prefisso *`Tax/`* nel nome della chiave.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
```

```
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>

    <Filter>
       <Prefix>Tax/</Prefix>
    </Filter>

    <Destination><Bucket>arn:aws:s3:::example-s3-bucket</Bucket></Destination>

  </Rule>
</ReplicationConfiguration>
```

Se specifichi l'elemento `Filter`, devi includere anche gli elementi `Priority` e `DeleteMarkerReplication`. In questo esempio, `Priority` non è rilevante perché è presente solo una regola.

Nella configurazione seguente il filtro specifica un prefisso e due tag. La regola si applica al sottoinsieme di oggetti con il prefisso della chiave e i tag specificati. Nello specifico, si applica all'oggetto con il prefisso *Tax/* nei nomi delle chiavi e i due tag specificati. `Priority` non si applica perché è presente soltanto una regola.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>

    <Filter>
        <And>
          <Prefix>Tax/</Prefix>
          <Tag>
             <Tag>
                <Key>tagA</Key>
                <Value>valueA</Value>
             </Tag>
          </Tag>
```

```
          <Tag>
             <Tag>
                <Key>tagB</Key>
                <Value>valueB</Value>
             </Tag>
          </Tag>
       </And>

    </Filter>

    <Destination><Bucket>arn:aws:s3:::example-s3-bucket</Bucket></Destination>

  </Rule>
</ReplicationConfiguration>
```

È possibile specificare una classe di storage per le repliche degli oggetti come illustrato di seguito:

```
<?xml version="1.0" encoding="UTF-8"?>

<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Destination>
       <Bucket>arn:aws:s3:::example-s3-bucket</Bucket>
       <StorageClass>storage-class</StorageClass>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

È possibile specificare qualsiasi classe di storage supportata da Amazon S3.

Example 2: Configurazione di replica con due regole

Example

Nella configurazione di replica seguente:

- Ogni regola filtra in base a un prefisso della chiave diverso, per cui ogni regola si applica a un sottoinsieme di oggetti diverso. In questo esempio, Amazon S3 replica gli oggetti con i nomi di chiave *Tax/doc1.pdf* e *Project/project1.txt*, ma non gli oggetti con il nome di chiave *PersonalDoc/documentA*.

- La priorità delle regole non è rilevante perché le regole si applicano a due insiemi di oggetti distinti. L'esempio successivo mostra cosa accade quando viene applicata la priorità delle regole.
- La seconda regola specifica una classe di archiviazione S3 Standard-IA per le repliche degli oggetti. Amazon S3 usa la classe di storage specificata per tali repliche.

```
<?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: Configurazione di replica con due regole con prefissi sovrapposti

In questa configurazione, le due regole specificano filtri con prefissi della chiave che si sovrappongono, *star/* e *starship/*. Entrambe le regole si applicano agli oggetti con il nome di chiave *starship-x*. In questo caso, Amazon S3 utilizza la priorità delle regole per determinare quale regola applicare. Più elevato è il numero, maggiore è la priorità.

```
<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: Procedure dettagliate di esempio

Per le procedure dettagliate di esempio, consulta Esempi di configurazione della replica in tempo reale.

Per ulteriori informazioni sulla struttura XML della configurazione di replica, consulta PutBucketReplicationAmazon Simple Storage Service API Reference.

Compatibilità con le versioni precedenti

La versione più recente del codice XML di configurazione di replica è V2. Le configurazioni di replica XML V2 sono quelle che contengono l'elemento `Filter` per le regole e le regole che specificano S3 Replication Time Control (S3 RTC).

Per visualizzare la versione della configurazione di replica, puoi utilizzare l'operazione API `GetBucketReplication`. Per ulteriori informazioni, consulta il riferimento GetBucketReplicationall'API di Amazon Simple Storage Service.

Per la compatibilità con le versioni precedenti, Amazon S3 continua a supportare la configurazione della replica XML V1. Se hai usato la configurazione di replica XML V1, tieni presente i problemi seguenti relativi alla compatibilità con le versioni precedenti:

- Il codice XML di configurazione di replica V2 include l'elemento `Filter` per le regole. Con l'elemento `Filter`, puoi specificare filtri di oggetti basati sul prefisso della chiave o sui tag dell'oggetto, oppure su entrambi gli elementi, per specificare gli oggetti a cui si applica la regola. Filtraggio supportato della configurazione di replica XML V1 in base solo al prefisso della chiave. In tal caso, aggiungi `Prefix` direttamente come elemento figlio dell'elemento `Rule`, come nell'esempio seguente.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Prefix>key-prefix</Prefix>
    <Destination><Bucket>arn:aws:s3:::example-s3-bucket</Bucket></Destination>

  </Rule>
</ReplicationConfiguration>
```

Per la compatibilità con le versioni precedenti, Amazon S3 continua a supportare la configurazione V1.

- Quando elimini un oggetto dal bucket di origine senza specificare un ID versione, Amazon S3 aggiunge un contrassegno di eliminazione. Se utilizzi il codice XML di configurazione di replica V1, Amazon S3 replica i contrassegni di eliminazione derivanti dalle operazioni dell'utente. In altre parole, Amazon S3 esegue la replica del contrassegno di eliminazione solo se un utente elimina un oggetto. Se un oggetto scaduto viene rimosso da Amazon S3 (come parte di un'operazione del ciclo di vita), Amazon S3 non replicherà il contrassegno di eliminazione.

  Nelle configurazioni di replica V2, puoi abilitare la replica dei marker di eliminazione per le regole. non-tag-based Per ulteriori informazioni, consulta Replica dei contrassegni di eliminazione tra i bucket.

Impostazione delle autorizzazioni per la replica in tempo reale

Quando si configura la replica live, è necessario acquisire le autorizzazioni necessarie come segue:

- Amazon S3 necessita delle autorizzazioni per replicare gli oggetti per tuo conto. Queste autorizzazioni vengono concesse creando un ruolo IAM e specificandolo nella configurazione di replica.
- Quando i bucket di origine e di destinazione non sono di proprietà degli stessi account, il proprietario del bucket di destinazione deve concedere al proprietario del bucket di origine le autorizzazioni per archiviare le repliche.

Argomenti

- Creazione di un ruolo IAM
- Concessione delle autorizzazioni quando i bucket di origine e di destinazione sono di proprietà di diversi Account AWS
- Concessione di autorizzazioni per le operazioni in batch S3
- Modifica del proprietario della replica
- Abilita la ricezione di oggetti replicati da un bucket di origine

Creazione di un ruolo IAM

Di default, tutte le risorse di Amazon S3, ossia bucket, oggetti e risorse secondarie correlate, sono private e solo il proprietario vi può accedere. Amazon S3 ha bisogno di autorizzazioni per leggere e replicare gli oggetti dal bucket di origine. Queste autorizzazioni vengono concesse creando un ruolo IAM e specificandolo nella configurazione di replica.

In questa sezione vengono illustrate la policy di trust e la policy di autorizzazione minima richiesta. Le procedure dettagliate di esempio forniscono step-by-step istruzioni per creare un ruolo IAM. Per ulteriori informazioni, consulta Esempi di configurazione della replica in tempo reale.

- Di seguito viene mostrata una policy di attendibilità in cui identifichi Amazon S3 come principale del servizio che può assumere il ruolo.

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

- Nell'esempio seguente viene illustrata una policy di attendibilità in cui identifichi Amazon S3 e Operazioni di batch S3 come principali del servizio. Ciò è utile quando crei un processo di replica in batch. Per ulteriori informazioni, consulta Creazione di un processo Batch Replication per una prima regola di replica o una nuova destinazione.

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

```
            "Action":"sts:AssumeRole"
        }
    ]
}
```

Per ulteriori informazioni sui ruoli IAM, consulta Ruoli IAM nella Guida per l'utente di IAM.

- Di seguito viene mostrata una policy di accesso in cui concedi al ruolo le autorizzazioni per eseguire attività di replica per tuo conto. Quando Amazon S3 assume il ruolo, dispone delle autorizzazioni che sono state specificate in questa policy. In questa politica, *example-s3-bucket1* è il bucket di origine e *example-s3-bucket2* è il nome del bucket di destinazione.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::example-s3-bucket1"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
         ],
         "Resource":[
            "arn:aws:s3:::example-s3-bucket1/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete",
            "s3:ReplicateTags"
         ],
```

```
            "Resource":"arn:aws:s3:::example-s3-bucket2/*"
        }
    ]
}
```

La policy di accesso concede le autorizzazioni per le seguenti operazioni:

- `s3:GetReplicationConfiguration` e `s3:ListBucket`— Autorizzazioni per queste azioni sul bucket *`example-s3-bucket1`* (il bucket di origine) consentono ad Amazon S3 di recuperare la configurazione di replica ed elencare il contenuto del bucket. (Il modello di autorizzazioni corrente richiede l'autorizzazione `s3:ListBucket` per l'accesso ai contrassegni di eliminazione.)
- `s3:GetObjectVersionForReplication` e `s3:GetObjectVersionAcl`: le autorizzazioni per queste operazioni concesse su tutti gli oggetti permettono ad Amazon S3 di ottenere una versione dell'oggetto specifica e la lista di controllo degli accessi (ACL) associata agli oggetti.

- `s3:ReplicateObject` e `s3:ReplicateDelete`: le autorizzazioni per queste operazioni sugli oggetti nel bucket di *`example-s3-bucket2`* (il bucket di destinazione) permettono ad Amazon S3 di replicare gli oggetti o eliminare i contrassegni nel bucket di destinazione. Per informazioni sui contrassegni di eliminazione, consulta la sezione Effetto delle operazioni di eliminazione sulla replica.

  > **Note**
  >
  > Le autorizzazioni per l'operazione `s3:ReplicateObject` nel bucket *`example-s3-bucket2`* (il bucket di destinazione) consentono anche la replica dei metadati, come i tag degli oggetti e le liste di controllo degli accessi. Pertanto non è necessario concedere esplicitamente l'autorizzazione per l'operazione `s3:ReplicateTags`.

- `s3:GetObjectVersionTagging`: le autorizzazioni per questa operazione sugli oggetti nel bucket *`example-s3-bucket1`* (bucket di origine) permettono ad Amazon S3 di leggere i tag degli oggetti per la replica. Per ulteriori informazioni, consulta Suddivisione in categorie dello storage utilizzando i tag. Se Amazon S3 non dispone di queste autorizzazioni, replica gli oggetti ma non i relativi tag.

Per un elenco delle azioni di Amazon S3, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.

> **Important**
>
> Il Account AWS proprietario del ruolo IAM deve disporre delle autorizzazioni per le azioni che concede al ruolo IAM.
> Supponiamo ad esempio che il bucket di origine contenga oggetti di proprietà di un altro Account AWS. Il proprietario degli oggetti deve concedere esplicitamente al proprietario del Account AWS ruolo IAM le autorizzazioni richieste tramite l'ACL dell'oggetto. In caso contrario, Amazon S3 non può accedere agli oggetti e la replica degli oggetti ha esito negativo. Per informazioni sulle autorizzazioni ACL, consulta la sezione Panoramica delle liste di controllo accessi (ACL).
> Le autorizzazioni descritte si riferiscono alla configurazione di replica minima. Se scegli di aggiungere configurazioni di replica facoltative, devi concedere ulteriori autorizzazioni ad Amazon S3.

Concessione delle autorizzazioni quando i bucket di origine e di destinazione sono di proprietà di diversi Account AWS

Quando i bucket di origine e di destinazione non sono di proprietà degli stessi account, il proprietario del bucket di destinazione deve aggiungere anche una policy di bucket per concedere al proprietario del bucket di origine le autorizzazioni per eseguire le operazioni di replica, come illustrato di seguito. In questa policy, *example-s3-bucket2* è il nome del bucket di destinazione.

> **Note**
>
> Il formato ARN del ruolo potrebbe apparire diverso. Se il ruolo è stato creato utilizzando la console, il formato ARN è. `arn:aws:iam::`*account-ID*`:role/service-role/`*role-name* Se il ruolo è stato creato utilizzando il AWS CLI, il formato ARN è. `arn:aws:iam::`*account-ID*`:role/`*role-name* Per ulteriori informazioni, consulta Ruoli IAM nella Guida per l'utente IAM.

```
{
   "Version":"2012-10-17",
   "Id":"PolicyForDestinationBucket",
   "Statement":[
      {
```

```
         "Sid":"Permissions on objects",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::SourceBucket-account-ID:role/service-role/source-
account-IAM-role"
         },
         "Action":[
            "s3:ReplicateDelete",
            "s3:ReplicateObject"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket2/*"
      },
      {
         "Sid":"Permissions on bucket",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::SourceBucket-account-ID:role/service-role/source-
account-IAM-role"
         },
         "Action": [
            "s3:List*",
            "s3:GetBucketVersioning",
            "s3:PutBucketVersioning"
         ],
         "Resource":"arn:aws:s3::example-s3-bucket2"
      }
   ]
}
```

Per vedere un esempio, consulta Configurazione della replica quando i bucket di origine e di destinazione sono di proprietà di account diversi.

In presenza di oggetti con tag nel bucket di origine, tenere in considerazione quanto segue:

- Se il proprietario del bucket di origine concede ad Amazon S3 l'autorizzazione per le operazioni `s3:GetObjectVersionTagging` e `s3:ReplicateTags` per replicare i tag degli oggetti (tramite il ruolo IAM), Amazon S3 replica i tag insieme agli oggetti. Per informazioni sul ruolo IAM, consulta Creazione di un ruolo IAM.
- Se il proprietario del bucket di destinazione non desidera replicare i tag, può aggiungere l'istruzione seguente alla policy del bucket di destinazione per rifiutare esplicitamente l'autorizzazione per l'operazione `s3:ReplicateTags`. In questa politica, *example-s3-bucket2*è il nome del bucket di destinazione.

```
...
   "Statement":[
      {
         "Effect":"Deny",
         "Principal":{
            "AWS":"arn:aws:iam::SourceBucket-account-id:role/service-role/source-account-IAM-role"
         },
         "Action":"s3:ReplicateTags",
         "Resource":"arn:aws:s3:::example-s3-bucket2/*"
      }
   ]
...
```

Concessione di autorizzazioni per le operazioni in batch S3

S3 Batch Replication fornisce un modo per replicare gli oggetti che esistevano già prima della configurazione della replica, gli oggetti replicati in precedenza e gli oggetti la cui replica è fallita. Quando crei la prima regola in una nuova configurazione di replica o quando aggiungi una nuova destinazione a una configurazione esistente tramite la AWS Management Console, hai la possibilità di creare un processo Batch Replication una tantum. Inoltre, puoi avviare Batch Replication per una configurazione di replica esistente creando un processo Batch Operations.

Per esempi di ruoli e policy IAM di Batch Replication, consulta Configurazione delle policy IAM per Batch Replication.

Modifica del proprietario della replica

Se Account AWS il bucket di origine e quello di destinazione sono diversi, puoi chiedere ad Amazon S3 di cambiare la proprietà della replica con quella proprietaria del bucket di Account AWS destinazione. Per ulteriori informazioni sulla sovrascrittura del proprietario, consulta Modifica del proprietario della replica.

Abilita la ricezione di oggetti replicati da un bucket di origine

È possibile generare rapidamente le policy necessarie per abilitare la ricezione di oggetti replicati da un bucket di origine tramite AWS Management Console.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckete, scegliere il bucket da utilizzare come bucket di destinazione.
4. Seleziona la scheda Gestione, quindi scorri verso il basso fino a Regole di replica.
5. Per Operazioni, scegliere Ricevi oggetti replicati.

   Segui le istruzioni e inserisci l' Account AWS ID dell'account bucket di origine e scegli Genera politiche. Ciò genererà una policy di bucket Amazon S3 e una policy di chiave KMS.
6. Per aggiungere questa policy alla policy del bucket esistente, scegli Applica le impostazioni oppure scegli Copia per copiare manualmente le modifiche.
7. (Facoltativo) Copia la AWS KMS politica nella policy chiave KMS desiderata sulla console. AWS Key Management Service

Esempi di configurazione della replica in tempo reale

Nei seguenti esempi viene mostrato come configurare la replica per i casi d'uso comuni.

> **Note**
>
> La replica in tempo reale fa riferimento alla replica nella stessa regione (SRR) e alla replica tra regioni (CRR). La replica in tempo reale non replica gli oggetti presenti nel bucket prima della configurazione della replica. Per replicare oggetti che esistevano prima di impostare la replica, utilizzate la replica su richiesta. Per sincronizzare i bucket e replicare oggetti esistenti su richiesta, vedere. Replica di oggetti esistenti

Questi esempi dimostrano come creare una configurazione di replica utilizzando la console Amazon S3 AWS Command Line Interface ,AWS CLI() e gli SDK AWS SDK for Java (AWS e vengono mostrati alcuni esempi). AWS SDK for .NET

Per informazioni sull'installazione e la configurazione di AWS CLI, consulta i seguenti argomenti nella Guida per l'utente.AWS Command Line Interface

- Installazione di AWS Command Line Interface
- Configurazione di AWS CLI: è necessario impostare almeno un profilo. Se stai esplorando scenari per più account, devi impostare due profili.

Per informazioni sugli SDK, consulta AWS AWS SDK for Java AWS e SDK for .NET.

> **Tip**
>
> Per un step-by-step tutorial che dimostra come utilizzare la replica live per replicare i dati, consulta Tutorial: Replica dei dati all'interno e tra l'uso di S3 Replication. Regioni AWS

Argomenti

Configurazione della replica per i bucket di origine e di destinazione di proprietà dello stesso account

La replica è la copia automatica e asincrona di oggetti tra bucket uguali o diversi. Regioni AWS Il processo replica gli oggetti appena creati e gli aggiornamenti degli oggetti da un bucket di origine a uno o più bucket di destinazione. Per ulteriori informazioni, consulta Panoramica sulla replica degli oggetti.

Quando si configura la replica, vengono aggiunte le regole di replica al bucket di origine. Le regole di replica definiscono gli oggetti del bucket di origine da replicare e i bucket di destinazione in cui vengono archiviati gli oggetti replicati. È possibile creare una regola per replicare tutti gli oggetti in un bucket o un sottoinsieme di oggetti con un prefisso di nome di chiave specifico, uno o più tag di oggetto o entrambi gli elementi. Un bucket di destinazione può trovarsi nello stesso del bucket Account AWS di origine o in un account diverso.

Se specifichi l'ID della versione dell'oggetto da eliminare, Amazon S3 elimina la versione dell'oggetto nel bucket di origine. Ma non replica l'eliminazione nel bucket di destinazione. In altre parole, non elimina la stessa versione dell'oggetto dal bucket di destinazione. Ciò permette di proteggere i dati da eliminazioni da parte di utenti malintenzionati.

Quando si aggiunge una regola di replica a un bucket, la regola viene abilitata per impostazione predefinita e pertanto inizia a funzionare non appena viene salvata.

In questo esempio viene configurata la replica per i bucket di origine e di destinazione di proprietà dello stesso Account AWS. Vengono forniti esempi per l'utilizzo della console Amazon S3, di AWS Command Line Interface (AWS CLI) e di and. AWS SDK for Java AWS SDK for .NET

Utilizzo della console S3

Per configurare una regola di replica quando il bucket di destinazione si trova nello stesso Account AWS del bucket di origine, segui questi passaggi.

Se il bucket di destinazione si trova in un account diverso rispetto al bucket di origine, è necessario aggiungere al bucket di destinazione una policy di bucket per concedere al proprietario dell'account del bucket di origine l'autorizzazione per replicare gli oggetti nel bucket di destinazione. Per ulteriori informazioni, consulta Concessione delle autorizzazioni quando i bucket di origine e di destinazione sono di proprietà di diversi Account AWS.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket scegli il nome del bucket desiderato.
4. Seleziona la scheda Gestione, scorri verso il basso fino a Regole di replica e quindi scegli Crea regola di replica.
5. Nella sezione Configurazione della regola di replica, in Nome della regola di replica, specifica il nome della regola per semplificarne l'identificazione in un secondo momento. Il nome è obbligatorio e deve essere univoco all'interno del bucket.
6. In Status (Stato), l'opzione Enabled (Abilitata) è selezionata per impostazione predefinita. Una regola abilitata inizia a funzionare non appena viene salvata. Se desideri abilitare la regola in un secondo momento, scegli Disabilitata.
7. Se il bucket dispone di regole di replica esistenti, viene chiesto di impostare una priorità per la regola. È necessario impostare una priorità per la regola per evitare i conflitti causati dagli oggetti inclusi nell'ambito di più regole. In caso di regole sovrapposte, Amazon S3 utilizza la priorità delle regole per determinare quale regola applicare. Più elevato è il numero, maggiore è la priorità. Per ulteriori informazioni sulla priorità delle regole, consulta Configurazione di replica.
8. In Bucket di origine sono disponibili le seguenti opzioni per l'impostazione dell'origine della replica:

- Per replicare l'intero bucket, scegli Apply to all objects in the bucket (Applica a tutti gli oggetti nel bucket).
- Per replicare tutti gli oggetti con lo stesso prefisso, scegli Limita l'ambito di questa regola utilizzando uno o più filtri. Ciò limita la replica a tutti gli oggetti con nomi che iniziano il prefisso specificato, ad esempio `pictures`. Immetti un prefisso nella casella Prefisso.

  > **Note**
  >
  > Se si immette un prefisso corrispondente al nome di una cartella, è necessario utilizzare / (barra) come ultimo carattere (ad esempio, `pictures/`).

- Per replicare tutti gli oggetti con uno o più tag oggetto, scegli Aggiungi tag e specifica la coppia chiave-valore nelle caselle. Per aggiungere un altro tag, ripetere la procedura. È possibile combinare un prefisso con i tag. Per ulteriori informazioni sui tag degli oggetti, consulta Suddivisione in categorie dello storage utilizzando i tag.

Il nuovo schema XML della configurazione della replica supporta il filtro basato su prefissi e tag e l'impostazione della priorità delle regole. Per ulteriori informazioni sul nuovo schema, consulta Compatibilità con le versioni precedenti. Per ulteriori informazioni sull'XML utilizzato con l'API Amazon S3 che funziona con l'interfaccia utente, consulta Configurazione di replica. Il nuovo schema è descritto come configurazione di replica XML V2.

9. In Destinazione, scegli il bucket in cui desideri che Amazon S3 esegui la replica degli oggetti.

   > **Note**
   >
   > Il numero di bucket di destinazione è limitato al numero di bucket Regioni AWS presenti in una determinata partizione. Una partizione è un raggruppamento di regioni. AWS attualmente ha tre partizioni: `aws` (Regioni standard), `aws-cn` (Regioni cinesi) e `aws-us-gov` (AWS GovCloud (US) Regioni). È possibile utilizzare le Service Quotas per richiedere un aumento del limite per i bucket di destinazione.

   - Per eseguire la replica in un periodo fisso nel tuo account, seleziona Scegli un bucket in questo account e digita o cerca i bucket di destinazione.

- Per eseguire la replica in uno o più bucket in un altro account Account AWS, scegli Specificare un bucket in un altro account e inserisci l'ID dell'account del bucket di destinazione e il nome del bucket.

  Se il bucket di destinazione si trova in un account diverso rispetto al bucket di origine, dovrai aggiungere ai bucket di destinazione una policy di bucket per concedere al proprietario dell'account del bucket di origine l'autorizzazione per replicare gli oggetti nei bucket di destinazione. Per ulteriori informazioni, consulta Concessione delle autorizzazioni quando i bucket di origine e di destinazione sono di proprietà di diversi Account AWS.

  Facoltativamente, se desideri standardizzare la proprietà dei nuovi oggetti nel bucket di destinazione, seleziona Assegna la proprietà degli oggetti al proprietario del bucket di destinazione. Per ulteriori informazioni su questa opzione, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

> **Note**
>
> Se la funzione Controllo delle versioni non è abilitata nel bucket di destinazione, viene visualizzato un messaggio di avviso contenente un pulsante Abilita Controllo delle versioni. Seleziona questo pulsante per abilitare la funzione Controllo delle versioni nel bucket.

10. Imposta un ruolo AWS Identity and Access Management (IAM) che Amazon S3 può assumere per replicare oggetti per tuo conto.

    Per impostare un ruolo IAM, nella sezione Ruolo IAM seleziona uno dei seguenti valori nell'elenco a discesa Ruolo IAM:

    - Consigliamo di scegliere Crea nuovo ruolo per fare in modo che Amazon S3 crei un nuovo ruolo IAM per l'utente. Quando salvi la regola, viene generata una nuova policy per il ruolo IAM corrispondente ai bucket di origine e di destinazione scelti.
    - Puoi decidere di utilizzare un ruolo IAM esistente. In tal caso, è necessario scegliere un ruolo che conceda ad Amazon S3 le autorizzazioni necessarie per la replica. Se questo ruolo non concede autorizzazioni sufficienti ad Amazon S3 per seguire la regola di replica, la replica non riesce.

> **Important**
>
> Quando si aggiunge una regola di replica a un bucket, si deve disporre dell'autorizzazione `iam:PassRole` per poter passare il ruolo IAM che concede le autorizzazioni di replica Amazon S3. Per ulteriori informazioni, consulta Concessione di autorizzazioni utente per il passaggio di un ruolo a un Servizio AWS nella Guida per l'utente di IAM.

11. Per replicare gli oggetti nel bucket di origine che sono crittografati con crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS), in Crittografia, seleziona Replica oggetti crittografati con. AWS KMS In Chiavi AWS KMS per crittografare gli oggetti di destinazione sono disponibili le chiavi di origine che consentono la replica da utilizzare. Tutte le chiavi KMS di origine sono incluse per impostazione predefinita. Per limitare la selezione delle chiavi KMS, puoi scegliere un alias o un ID chiave.

    Gli oggetti crittografati da quelli non selezionati AWS KMS keys non vengono replicati. Viene scelta una chiave KMS o un gruppo di chiavi KMS, ma se lo desideri puoi scegliere le chiavi KMS. Per informazioni sull'utilizzo AWS KMS con la replica, vedere. Replica di oggetti crittografati (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS)

    > **Important**
    >
    > Quando si replicano oggetti crittografati con AWS KMS, la frequenza di AWS KMS richiesta raddoppia nella regione di origine e aumenta nella regione di destinazione dello stesso importo. L'aumento delle frequenze di chiamata è dovuto al modo in cui i dati AWS KMS vengono ricrittografati utilizzando la chiave KMS definita per la regione di destinazione della replica. AWS KMS ha una quota di frequenza di richiesta per account chiamante per regione. Per informazioni sulle quote predefinite, consulta la sezione Quote di AWS KMS - richieste al secondo: variabili nella Guida per gli sviluppatori di AWS Key Management Service .
    > Se la tua attuale frequenza di richieste di PUT oggetti Amazon S3 durante la replica è superiore alla metà del limite di AWS KMS velocità predefinito per il tuo account, ti consigliamo di richiedere un aumento della quota di frequenza delle AWS KMS richieste. Per richiedere un incremento, invia una richiesta tramite il AWS Support Center nella sezione Contatti. Ad esempio, supponiamo che la tua attuale frequenza di richieste di PUT oggetti sia di 1.000 richieste al secondo e che tu le utilizzi per AWS

KMS crittografare gli oggetti. In questo caso, ti consigliamo di chiedere AWS Support di aumentare il limite di AWS KMS frequenza a 2.500 richieste al secondo, sia nella regione di origine che in quella di destinazione (se diversa), per assicurarti che non vi siano limitazioni. AWS KMS
Per visualizzare la frequenza delle richieste di PUT oggetti nel bucket di origine, consulta i parametri di CloudWatch richiesta di Amazon per Amazon S3. `PutRequests` Per informazioni sulla visualizzazione CloudWatch dei parametri, consulta. Utilizzo della console S3

Se hai scelto di replicare gli oggetti crittografati con AWS KMS, procedi come segue:

- In AWS KMS key per crittografare gli oggetti di destinazione, specifica la tua chiave KMS in uno dei seguenti modi:
  - Per effettuare una selezione in un elenco di chiavi KMS disponibili, seleziona Scegli tra le chiavi AWS KMS keys e quindi scegli una chiave KMS dell'elenco delle chiavi disponibili.

    In questo elenco vengono visualizzate sia la chiave Chiave gestita da AWS (`aws/s3`) che quella gestita dal cliente. Per ulteriori informazioni sulle chiavi gestite dal cliente, consulta Chiavi gestite dal cliente e chiavi AWS nella Guida per gli sviluppatori di AWS Key Management Service .
  - Per inserire il nome della risorsa Amazon (ARN) della chiave KMS, scegli Inserisci ARN AWS KMS key e specifica l'ARN della chiave KMS nel campo visualizzato. In questo modo vengono crittografate le repliche nel bucket di destinazione. Puoi trovare l'ARN per la tua chiave KMS nella console IAM, sotto Chiavi di crittografia.
  - Per creare una nuova chiave gestita dal cliente nella AWS KMS console, scegli Crea una chiave KMS.

    Per ulteriori informazioni sulla creazione di una AWS KMS key, consulta Creating keys nella AWS Key Management Service Developer Guide.

> **Important**
>
> Puoi usare solo chiavi KMS abilitate nello Regione AWS stesso bucket. Quando selezioni Scegli tra le chiavi KMS, la console S3 elenca solo 100 chiavi KMS per regione. Se hai oltre 100chiavi KMS nella stessa regione, puoi vedere solo i primi

le prime 100 nella console S3. Per utilizzare una chiave KMS non elencata nella console, seleziona Inserisci ARN AWS KMS key e specifica l'ARN della chiave KMS.
Quando utilizzi una chiave KMS AWS KMS key per la crittografia lato server in Amazon S3, devi scegliere una chiave KMS di crittografia simmetrica. Amazon S3 supporta solo chiavi KMS di crittografia simmetriche e non chiavi KMS asimmetriche. Per ulteriori informazioni, consulta Identificazione delle chiavi KMS simmetriche e asimmetriche nella Guida per gli sviluppatori di AWS Key Management Service .

Per ulteriori informazioni sulla creazione di una AWS KMS key, consulta Creating keys nella Developer Guide.AWS Key Management Service Per ulteriori informazioni sull'utilizzo AWS KMS con Amazon S3, consulta. Utilizzo della crittografia lato server con chiavi (SSE-KMS) AWS KMS

12. In Classe di storage di destinazione, per replicare i dati in una classe di archiviazione specifica nel bucket di destinazione, seleziona Modifica classe di archiviazione per gli oggetti replicati. Scegli quindi la classe di storage che desideri utilizzare per gli oggetti replicati nel bucket di destinazione. Se non selezioni questa opzione, la classe di storage per gli oggetti replicati sarà la stessa degli oggetti originali.
13. Durante l'impostazione dei valori in Opzioni di replica aggiuntive, sono disponibili le seguenti opzioni aggiuntive:

    - Se desideri abilitare la funzionalità di controllo del tempo di replica di S3 (S3 RTC) nella configurazione della replica, seleziona Controllo del tempo di replica (RTC). Per ulteriori informazioni su questa opzione, consulta Rispetto dei requisiti di conformità utilizzando S3 Replication Time Control (S3 RTC).
    - Se desideri abilitare i parametri di replica S3 nella configurazione di replica, seleziona Replication metrics and events (Parametri ed eventi di replica). Per ulteriori informazioni, consulta Monitoraggio dell'avanzamento con le metriche di replica e le notifiche eventi di Amazon S3.
    - Se desideri abilitare la replica del contrassegno di eliminazione nella configurazione di replica, seleziona Replica del contrassegno di eliminazione. Per ulteriori informazioni, consulta Replica dei contrassegni di eliminazione tra i bucket.
    - Se desideri abilitare la sincronizzazione delle modifiche alla replica di Amazon S3 nella configurazione di replica, seleziona Sincronizzazione delle modifiche alla replica. Per ulteriori

informazioni, consulta Replica delle modifiche ai metadati con la sincronizzazione delle modifiche alla replica Amazon S3.

> **Note**
>
> Quando si utilizzano i parametri di replica S3 RTC o S3, si applicano costi aggiuntivi.

14. Per terminare, seleziona Salva.
15. Dopo aver salvato la regola, potrai modificare, abilitare, disabilitare o eliminare la regola selezionando la regola e scegliendo Modifica regola.

Usando il AWS CLI

Per utilizzare il AWS CLI per impostare la replica quando i bucket di origine e di destinazione sono di proprietà dello stesso Account AWS, procedi come segue:

- Creazione dei bucket di origine e di destinazione
- Abilitazione del controllo delle versioni sui bucket
- Creazione di un ruolo IAM che permette ad Amazon S3 di replicare gli oggetti
- Aggiunta della configurazione di replica al bucket di origine

Per verificare l'impostazione, testarla.

Per impostare la replica quando i bucket di origine e di destinazione sono di proprietà dello stesso Account AWS

1. Impostare un profilo di credenziali per la AWS CLI. utilizzando il nome profilo `acctA`. Per informazioni sull'impostazione di profili con credenziali, consulta Profili denominati nella Guida per l'utente di AWS Command Line Interface .

> **Important**
>
> Il profilo utilizzato per questo esercizio deve disporre delle autorizzazioni necessarie. Ad esempio, nella configurazione di replica dovrai specificare il ruolo IAM che Amazon S3 può assumere. Puoi effettuare questa operazione solo se il profilo che utilizzi dispone dell'autorizzazione `iam:PassRole`. Per ulteriori informazioni, consulta Concessione di autorizzazioni utente per il passaggio di un ruolo a un servizio AWS nella Guida per

l'utente di IAM. Se utilizzi le credenziali di amministratore per creare un profilo con nome, puoi eseguire tutte le attività.

2. Crea un bucket di *source* e abilita la funzione di controllo delle versioni. Il codice seguente crea un bucket di *source* nella regione Stati Uniti orientali (Virginia settentrionale) (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. Crea un bucket di *destination* e abilita la funzione di controllo delle versioni. Il codice seguente crea un bucket di *destination* nella regione Stati Uniti occidentali (Oregon) (us-west-2).

> **Note**
>
> Per configurare la configurazione di replica quando entrambi i bucket di origine e di destinazione si trovano nello stesso profilo Account AWS, si utilizza lo stesso profilo. Questo esempio usa `acctA`. Per testare la configurazione di replica quando i bucket sono di proprietà di diversi Account AWS, è necessario specificare profili diversi per ciascuno. Questo esempio utilizza il profilo `acctB` per il bucket di destinazione.

```
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. Creare un ruolo IAM. Specifica questo ruolo nella configurazione di replica che aggiungi al bucket *source* in un secondo momento. Amazon S3 assume questo ruolo per replicare gli oggetti per tuo conto. Il ruolo IAM si crea in due fasi:

   - Creare un ruolo.
   - Collegare una policy di autorizzazione al ruolo.

   a. Crea il ruolo IAM.

      i. Copiare la seguente policy di attendibilità e salvarla in un file denominato `s3-role-trust-policy.json` nella directory corrente sul computer locale. Questa policy concede ad Amazon S3 le autorizzazioni ai principali del servizio per assumere il ruolo.

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

      ii. Per creare un ruolo, eseguire il comando seguente.

```
$ aws iam create-role \
--role-name replicationRole \
--assume-role-policy-document file://s3-role-trust-policy.json  \
--profile acctA
```

   b. Collegare una policy di autorizzazione al ruolo.

      i. Copiare la seguente policy di autorizzazioni e salvarla in un file denominato `s3-role-permissions-policy.json` nella directory corrente sul computer locale. Questa

policy di accesso concede le autorizzazioni per varie operazioni su oggetti e bucket 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. Eseguire il comando seguente per creare una policy e collegarla al ruolo.

```
$ aws iam put-role-policy \
--role-name replicationRole \
```

```
--policy-document file://s3-role-permissions-policy.json \
--policy-name replicationRolePolicy \
--profile acctA
```

5. Aggiungi la configurazione di replica al bucket di *source*.

   a. Sebbene l'API Amazon S3 richieda la configurazione di replica come XML, AWS CLI richiede che tu specifichi la configurazione di replica come JSON. Salvare il seguente JSON in un file denominato `replication.json` nella directory locale sul computer in uso.

```
{
  "Role": "IAM-role-ARN",
  "Rules": [
    {
      "Status": "Enabled",
      "Priority": 1,
      "DeleteMarkerReplication": { "Status": "Disabled" },
      "Filter" : { "Prefix": "Tax"},
      "Destination": {
        "Bucket": "arn:aws:s3:::destination-bucket"
      }
    }
  ]
}
```

   b. Aggiorna il JSON fornendo i valori per il *destination-bucket* e il *IAM-role-ARN*. Salvare le modifiche.

   c. Eseguire il comando seguente per aggiungere la configurazione di replica al bucket di origine. Assicurati di fornire il nome del bucket di *source*.

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket source \
--profile acctA
```

   Per recuperare la configurazione di replica, utilizzare il comando `get-bucket-replication`.

```
$ aws s3api get-bucket-replication \
--bucket source \
--profile acctA
```

6. Verifica la configurazione nella console di Amazon S3:

   a. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

   b. Nel bucket di *source*, crea una cartella denominata `Tax`.

   c. Aggiungi oggetti di esempio alla cartella `Tax` del bucket di *source*.

   > **Note**
   >
   > Il tempo richiesto da Amazon S3 per la replica di un oggetto dipende dalle dimensioni dell'oggetto. Per informazioni su come visualizzare lo stato della replica, consulta la sezione Ottenimento delle informazioni sullo stato della replica.

   Nel bucket di *destination*, verifica quanto segue:

   - Amazon S3 ha replicato gli oggetti.
   - Nelle proprietà dell'oggetto, Stato di replica è impostato su `Replica` (che identifica l'oggetto come replica).
   - Nelle proprietà dell'oggetto, nella sezione delle autorizzazioni non viene visualizzata alcuna autorizzazione. Questo significa che la replica appartiene ancora al proprietario del bucket di *source* e il proprietario del bucket di *destination* non possiede alcuna autorizzazione per la replica dell'oggetto. È possibile aggiungere una configurazione facoltativa per indicare ad Amazon S3 di cambiare la proprietà della replica. Per vedere un esempio, consulta Come modificare il proprietario della replica.

Utilizzo degli SDK AWS

Utilizzate i seguenti esempi di codice per aggiungere una configurazione di replica a un bucket con, rispettivamente AWS SDK for Java . AWS SDK for .NET

Java

L'esempio seguente aggiunge una configurazione di replica a un bucket e successivamente recupera e verifica la configurazione. Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
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#

Il seguente esempio di AWS SDK for .NET codice aggiunge una configurazione di replica a un bucket e quindi la recupera. Per utilizzare questo codice, fornisci i nomi dei bucket e l'Amazon Resource Name (ARN) per il ruolo IAM. Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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);
            }
        }
    }
}
```

Configurazione della replica quando i bucket di origine e di destinazione sono di proprietà di account diversi

L'impostazione della replica quando i bucket di *origine* e di *destinazione* sono di proprietà di diversi Account AWS è simile all'impostazione della replica quando entrambi i bucket sono di proprietà dello stesso account. L'unica differenza è che il proprietario del bucket *di destinazione* deve concedere al proprietario del bucket *di origine* l'autorizzazione per replicare gli oggetti aggiungendo una policy di bucket.

Per ulteriori informazioni sulla configurazione della replica utilizzando la crittografia lato server con AWS Key Management Service in scenari che coinvolgono molteplici account, consulta Concessione di autorizzazioni aggiuntive per scenari multi-account.

Configurare la replica quando i bucket di origine e di destinazione sono di proprietà di diversi Account AWS

1. In questo esempio, si creano i bucket di *origine* e di *destinazione* in due diversi. Account AWSÈ necessario impostare due profili di credenziali per AWS CLI (in questo esempio, utilizziamo `acctA` e `acctB` per i nomi dei profili). Per ulteriori informazioni sull'impostazione di profili con credenziali, consulta Profili denominati nella Guida per l'utente di AWS Command Line Interface .
2. Segui le step-by-step istruzioni riportate di seguito Configurazione per i bucket nello stesso account con le seguenti modifiche:
   - Per tutti AWS CLI i comandi relativi alle attività del bucket di *origine* (per creare il bucket di *origine*, abilitare il controllo delle versioni e creare il ruolo IAM), utilizza il profilo. `acctA` Utilizzare il profilo `acctB` per creare il bucket *di destinazione*.
   - Assicurarsi che la policy di autorizzazione specifichi i bucket *di origine* e *di destinazione* creati per questo esempio.
3. Nella console, aggiungere la seguente policy di bucket al bucket *di destinazione* per consentire al proprietario del bucket *di origine* di replicare gli oggetti. *Assicurati di modificare la policy fornendo l' Account AWS ID del proprietario del bucket di origine e il nome del bucket di destinazione.*

   > **Note**
   >
   > Per utilizzare il seguente esempio, sostituisci *user input placeholders* con le tue informazioni. Sostituisci *DOC-EXAMPLE-BUCKET* con il nome del tuo bucket di destinazione. Sostituisci *source-bucket-acct-id:role/service-role/source-acct-iam-Role* con il ruolo che stai utilizzando per questa configurazione di replica. Se hai creato il ruolo di servizio IAM manualmente, imposta il percorso del ruolo come `role/service-role/`, come mostrato nel seguente esempio di policy. Per ulteriori informazioni, consulta ARN IAM nella Guida per l'utente di 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"
      }
   ]
}
```

Seleziona il bucket e aggiungi la relativa policy. Per istruzioni, consulta Aggiunta di una policy di bucket utilizzando la console di Amazon S3.

Per impostazione predefinita, nella replica, il proprietario dell'oggetto di origine possiede anche la replica stessa. Quando i bucket di origine e di destinazione sono di proprietà di diversi, è possibile aggiungere impostazioni di configurazione opzionali per modificare la proprietà della replica con quella proprietaria dei bucket di destinazione. Account AWS Account AWS Ciò include la concessione dell'autorizzazione `ObjectOwnerOverrideToBucketOwner`. Per ulteriori informazioni, consulta Modifica del proprietario della replica.

Modifica del proprietario della replica

Nella replica, il proprietario dell'oggetto di origine possiede anche la replica per impostazione predefinita. Quando i bucket di origine e di destinazione sono di proprietà di diversi Account AWS e desideri modificare la proprietà della replica con Account AWS quella proprietaria dei bucket di destinazione, puoi aggiungere impostazioni di configurazione opzionali per modificare la proprietà della replica a quella proprietaria dei bucket di destinazione. Account AWS Ad esempio, è possibile eseguire questa operazione per limitare l'accesso alle repliche degli oggetti. Questa operazione viene definita sostituzione del proprietario della configurazione della replica. Per ulteriori informazioni

sull'opzione di sovrascrittura del proprietario, consulta la sezione Aggiunta dell'opzione di sostituzione del proprietario alla configurazione della replica. Per ulteriori informazioni sull'impostazione della configurazione della replica, consulta la sezione Panoramica sulla replica degli oggetti.

Per configurare la sostituzione del proprietario, procedi come segue:

- Aggiungi l'opzione di sostituzione del proprietario alla configurazione della replica per indicare ad Amazon S3 di modificare la proprietà della replica.
- Concedi ad Amazon S3 le autorizzazioni per modificare la proprietà della replica.
- Aggiungi l'autorizzazione alla policy del bucket di destinazione per consentire la modifica della proprietà della replica. Ciò consente al proprietario dei bucket di destinazione di accettare la proprietà delle repliche dell'oggetto.

Per ulteriori informazioni, consulta Aggiunta dell'opzione di sostituzione del proprietario alla configurazione della replica. Per un esempio pratico con istruzioni, vedere. step-by-step Come modificare il proprietario della replica

Impostazione proprietario del bucket applicato per Object Ownership

Quando utilizzi la replica di Amazon S3 e i bucket di origine e di destinazione sono di proprietà di diversi Account AWS, il proprietario del bucket di destinazione può disabilitare gli ACL (con l'impostazione imposta dal proprietario del bucket per Object Ownership) per modificare la proprietà della replica con quella proprietaria del bucket di destinazione. Account AWS Questa impostazione imita il comportamento di sovrascrittura del proprietario esistente senza la necessità di un'autorizzazione `s3:ObjectOwnerOverrideToBucketOwner`. Tutti gli oggetti replicati nel bucket di destinazione con l'impostazione proprietario del bucket applicato sono di proprietà del proprietario del bucket di destinazione. Per ulteriori informazioni su Object Ownership, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

Aggiunta dell'opzione di sostituzione del proprietario alla configurazione della replica

> **Warning**
>
> Aggiungi l'opzione Override del proprietario solo quando i bucket di origine e di destinazione sono di proprietà di diversi. Account AWS Amazon S3 non controlla se i bucket sono di proprietà dello stesso account o di account diversi. Se aggiungi l'override del proprietario quando entrambi i bucket sono di proprietà dello stesso Account AWS, Amazon S3 applica l'override del proprietario. Concede le autorizzazioni complete al proprietario del bucket di

destinazione e non replica gli aggiornamenti successivi nella lista di controllo degli accessi (ACL) dell'oggetto di origine. Il proprietario della replica può modificare direttamente l'ACL associata a una replica con una richiesta PUT ACL, ma non tramite replica.

Per specificare l'opzione di sostituzione del proprietario, aggiungi quanto segue all'elemento `Destination`:

- L'elemento `AccessControlTranslation`, che indica ad Amazon S3 di modificare la proprietà della replica
- L'`Account`elemento, che specifica il proprietario del Account AWS bucket di destinazione

```
<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 seguente configurazione della replica di esempio indica ad Amazon S3 di replicare gli oggetti con il prefisso della chiave `Tax` nel bucket di destinazione e di modificare la proprietà delle repliche.

```
<?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>
```

Concessione ad Amazon S3 dell'autorizzazione per modificare la proprietà della replica

Concedi ad Amazon S3 le autorizzazioni per modificare la proprietà della replica aggiungendo l'autorizzazione per l'operazione `s3:ObjectOwnerOverrideToBucketOwner` alla policy di autorizzazione associata al ruolo IAM. Questo è il ruolo IAM specificato nella configurazione della replica che consente ad Amazon S3 di acquisire e replicare gli oggetti per conto tuo.

```
...
{
    "Effect":"Allow",
         "Action":[
       "s3:ObjectOwnerOverrideToBucketOwner"
    ],
    "Resource":"arn:aws:s3:::destination-bucket/*"
}
...
```

Aggiunta dell'autorizzazione alla policy del bucket di destinazione per consentire la modifica della proprietà della replica

Il proprietario del bucket di destinazione deve concedere al proprietario del bucket di origine l'autorizzazione necessaria per modificare la proprietà della replica. Il proprietario del bucket di destinazione concede al proprietario del bucket di origine l'autorizzazione per l'operazione `s3:ObjectOwnerOverrideToBucketOwner`. Ciò consente al proprietario del bucket di destinazione di accettare la proprietà delle repliche dell'oggetto. La seguente istruzione della policy del bucket di esempio mostra come fare:

```
...
{
    "Sid":"1",
    "Effect":"Allow",
```

```
    "Principal":{"AWS":"source-bucket-account-id"},
    "Action":["s3:ObjectOwnerOverrideToBucketOwner"],
    "Resource":"arn:aws:s3:::destination-bucket/*"
}
...
```

Ulteriori considerazioni

Quando configuri l'opzione di sostituzione del proprietario, si applicano le seguenti considerazioni:

- Per impostazione predefinita, il proprietario dell'oggetto di origine possiede anche la replica. Amazon S3 replica la versione dell'oggetto e l'ACL associata.

  Se aggiungi la sostituzione del proprietario, Amazon S3 replica solo la versione dell'oggetto, non l'ACL. Inoltre, Amazon S3 non replica le modifiche successive all'ACL dell'oggetto di origine. Amazon S3 imposta l'ACL sulla replica che concede il controllo completo al proprietario del bucket di destinazione.
- Quando aggiorni una configurazione di replica per abilitare o disabilitare la sostituzione del proprietario, si verifica quanto segue.

  - Se aggiungi l'opzione di sostituzione del proprietario alla configurazione della replica:

    Quando replica una versione dell'oggetto, Amazon S3 elimina l'ACL associata all'oggetto di origine e imposta l'ACL sulla replica concedendo il controllo completo al proprietario del bucket di destinazione. Non replica le modifiche successive all'ACL dell'oggetto di origine. Tuttavia, questa modifica dell'ACL non si applica alle versioni dell'oggetto che sono state replicate prima di impostare l'opzione di sostituzione proprietario. Gli aggiornamenti all'ACL negli oggetti di origine che sono stati replicati prima che fosse impostata la sostituzione del proprietario continuano a essere replicati in quanto l'oggetto e le relative repliche continuano ad avere lo stesso proprietario.
  - Se rimuovi l'opzione di sostituzione del proprietario dalla configurazione della replica:

    Amazon S3 replica i nuovi oggetti presenti nel bucket di origine e le ACL associate ai bucket di destinazione. Per gli oggetti che sono stati replicati prima della rimozione della sostituzione del proprietario, Amazon S3 non replica le ACL perché rimane valida la modifica della proprietà dell'oggetto apportata da Amazon S3. In altre parole, le ACL associate alla versione dell'oggetto replicata quando la sostituzione del proprietario era impostata continuano a non essere replicate.

Come modificare il proprietario della replica

Quando i bucket di origine e di destinazione in una configurazione di replica sono di proprietà di diversi Account AWS, puoi dire ad Amazon S3 di cambiare la proprietà della replica con quella proprietaria Account AWS del bucket di destinazione. Questo esempio spiega come utilizzare la console Amazon S3 e modificare la proprietà delle AWS CLI repliche. Per ulteriori informazioni, consulta Modifica del proprietario della replica.

> **Note**
>
> Quando si utilizza la replica S3 e i bucket di origine e di destinazione sono di proprietà di diversi Account AWS, il proprietario del bucket di destinazione può disabilitare gli ACL (con l'impostazione imposta dal proprietario del bucket per Object Ownership) per modificare la proprietà della replica con quella proprietaria del bucket di destinazione. Account AWS Questa impostazione imita il comportamento di sovrascrittura del proprietario esistente senza la necessità di un'autorizzazione `s3:ObjectOwnerOverrideToBucketOwner`. Tutti gli oggetti replicati nel bucket di destinazione con l'impostazione proprietario del bucket applicato sono di proprietà del proprietario del bucket di destinazione. Per ulteriori informazioni su Object Ownership, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

Per ulteriori informazioni sulla configurazione della replica utilizzando la crittografia lato server negli scenari tra più account, consulta. AWS Key Management Service Concessione di autorizzazioni aggiuntive per scenari multi-account

Utilizzo della console S3

Per istruzioni, vedere. step-by-step Configurazione della replica per i bucket di origine e di destinazione di proprietà dello stesso account Questo argomento fornisce istruzioni per impostare la configurazione di replica quando i bucket sono di proprietà uguale o diversa. Account AWS

Usare il AWS CLI

Per modificare la proprietà delle repliche utilizzando AWS CLI, è necessario creare bucket, abilitare il controllo delle versioni sui bucket, creare un ruolo IAM che autorizzi Amazon S3 a replicare oggetti e aggiungere la configurazione di replica al bucket di origine. Nella configurazione di replica si indica ad Amazon S3 di modificare il proprietario della replica. Si esegue inoltre il test della configurazione.

Per modificare la proprietà della replica quando i bucket di origine e di destinazione sono di proprietà di diversi () Account AWSAWS CLI

1. In questo esempio, si creano i bucket di *origine* e di *destinazione* in due diversi. Account AWS Configurali AWS CLI con due profili denominati. Questo esempio utilizza i profili denominati rispettivamente `acctA` e `acctB`. Per ulteriori informazioni sull'impostazione di profili con credenziali, consulta Profili denominati nella Guida per l'utente di AWS Command Line Interface .

   > **Important**
   >
   > I profili utilizzati per questo esercizio devono disporre delle autorizzazioni necessarie. Ad esempio, nella configurazione di replica dovrai specificare il ruolo IAM che Amazon S3 può assumere. Puoi effettuare questa operazione solo se il profilo che utilizzi dispone dell'autorizzazione `iam:PassRole`. Se utilizzi le credenziali dell'utente amministratore per creare un profilo denominato, allora puoi eseguire tutte le attività. Per ulteriori informazioni, consulta Garantire a un utente le autorizzazioni per passare un ruolo a un AWS servizio nella Guida per l'utente IAM.

   Assicurarsi che questi profili abbiano le autorizzazioni necessarie. Ad esempio, la configurazione di replica include un ruolo IAM che Amazon S3 può assumere. Il profilo denominato utilizzato per allegare tale configurazione a un bucket può eseguire questa operazione solo se dispone dell'autorizzazione `iam:PassRole`. Se si specificano le credenziali dell'utente amministratore durante la creazione di questi profili denominati, vengono fornite tutte le autorizzazioni. Per ulteriori informazioni, consulta Garantire a un utente le autorizzazioni per passare un ruolo a un AWS servizio nella Guida per l'utente IAM.

2. Creare il bucket di *origine* e abilitare la funzione Controllo delle versioni. Questo esempio crea il bucket *source* nella regione Stati Uniti orientali (Virginia settentrionale) (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. Creare un bucket di *destinazione* e abilitare la funzione Controllo delle versioni. Questo esempio crea il bucket *destination* nella regione Stati Uniti occidentali (Oregon) (us-west-2). Utilizza un profilo Account AWS diverso da quello che hai utilizzato per il bucket di *fonte*.

```
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. Devi aggiungere l'autorizzazione alla policy del bucket di *destinazione* per consentire la modifica della proprietà della replica.

   a. Salvare la seguente policy su *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. Inserire la policy di cui sopra al bucket di *destinazione*:

```
aws s3api put-bucket-policy --region $ {destination_region} --
bucket $ {destination} --policy file://destination_bucket_policy.json
```

5. Creare un ruolo IAM. Specifica questo ruolo nella configurazione di replica che aggiungi al bucket *source* in un secondo momento. Amazon S3 assume questo ruolo per replicare gli oggetti per tuo conto. Il ruolo IAM si crea in due fasi:

   - Creare un ruolo.
   - Collegare una policy di autorizzazione al ruolo.

   a. Creare un ruolo IAM.

      i. Copiare la seguente policy di attendibilità e salvarla in un file denominato `s3-role-trust-policy.json` nella directory corrente sul computer locale. Questa policy concede ad Amazon S3 le autorizzazioni per assumere il ruolo.

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

      ii. Esegui il AWS CLI comando seguente per creare un ruolo.

```
$ aws iam create-role \
--role-name replicationRole \
--assume-role-policy-document file://s3-role-trust-policy.json  \
--profile acctA
```

b. Collegare una policy di autorizzazione al ruolo.

i. Copiare la seguente policy di autorizzazioni e salvarla in un file denominato `s3-role-perm-pol-changeowner.json` nella directory corrente sul computer locale. Questa policy di accesso concede le autorizzazioni per varie operazioni su oggetti e bucket Amazon S3. Nelle fasi che seguono si crea un ruolo IAM e si collega questa policy al ruolo.

```
{
   "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. Per creare una policy e collegarla al ruolo, eseguire questo comando.

```
$ aws iam put-role-policy \
--role-name replicationRole \
--policy-document file://s3-role-perm-pol-changeowner.json \
--policy-name replicationRolechangeownerPolicy \
--profile acctA
```

6. Aggiungere una configurazione di replica al bucket di origine.

a. È AWS CLI necessario specificare la configurazione di replica come JSON. Salvare il seguente JSON in un file denominato `replication.json` nella directory corrente sul computer locale. Nella configurazione, l'aggiunta di `AccessControlTranslation` per indicare la modifica nella proprietà della replica.

```
{
   "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. Modificare il JSON fornendo i valori per l'ID account del proprietario del bucket di *destinazione* e l'*ARN del ruolo IAM (IAM-role-ARN)*. Salvare le modifiche.

c. Per aggiungere la configurazione di replica al bucket di origine, eseguire il seguente comando. Fornire il nome del bucket di *origine*.

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket source \
--profile acctA
```

7. Verifica la proprietà della replica nella console di Amazon S3.

a. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

b. Aggiungere oggetti al bucket *di origine* . *Verifica che il bucket di destinazione contenga le repliche degli oggetti e che la proprietà delle repliche sia passata a Account AWS quella proprietaria del bucket di destinazione.*

Utilizzo degli SDK AWS

Per un esempio di codice per l'aggiunta della configurazione di replica, consulta Utilizzo degli SDK AWS. La configurazione della replica deve essere modificata di conseguenza. Per informazioni concettuali, consulta Modifica del proprietario della replica.

Rispetto dei requisiti di conformità utilizzando S3 Replication Time Control (S3 RTC)

Amazon S3 Replication Time Control (S3 RTC) permette di soddisfare i requisiti aziendali o di conformità per la replica dei dati fornendo visibilità sui tempi di replica di Amazon S3. S3 RTC replica la maggior parte degli oggetti caricati in Amazon S3 in pochi secondi e il 99,99% di tali oggetti entro 15 minuti.

Per impostazione predefinita, S3 RTC include i parametri di replica di S3 e le notifiche di eventi di Amazon S3, che puoi utilizzare per monitorare il numero totale di operazioni API S3 in attesa di replica, la dimensione totale degli oggetti in attesa di replica e il tempo massimo di replica. Puoi abilitare i parametri di replica indipendentemente da S3 RTC. Per ulteriori informazioni, consulta Monitoraggio dell'avanzamento con i parametri di replica. Inoltre, S3 RTC fornisce eventi `OperationMissedThreshold` e `OperationReplicatedAfterThreshold` che notificano al proprietario del bucket se la replica dell'oggetto supera o continua dopo la soglia di 15 minuti.

Con S3 RTC, gli eventi Amazon S3 possono inviarti una notifica nel caso raro in cui gli oggetti non vengano replicati entro 15 minuti e quando tali oggetti vengono replicati una volta superata la soglia di 15 minuti. Gli eventi Amazon S3 sono disponibili tramite Amazon SQS, Amazon SNS o. AWS Lambda Per ulteriori informazioni, consulta the section called “Notifiche di eventi Amazon S3”.

Argomenti

Controllo del tempo di replica S3

Puoi iniziare a utilizzare S3 Replication Time Control (S3 RTC) con una regola di replica nuova o esistente. Puoi decidere di applicare la regola di replica a un intero bucket S3 o a oggetti Amazon S3 con un prefisso o un tag specifico. Quando si attiva S3 RTC, i parametri di replica vengono abilitati anche nella regola di replica.

Se usi la versione più recente della configurazione di replica (ossia se specifichi l'elemento `Filter` in una regola di configurazione di replica), Amazon S3 non replica automaticamente il contrassegno di eliminazione. Tuttavia, è possibile aggiungere la replica dei marker di eliminazione alle regole. non-tag-based

> **Note**
>
> I parametri di replica vengono fatturati alla stessa tariffa dei parametri personalizzati di Amazon CloudWatch . Per informazioni, consulta i CloudWatchprezzi di Amazon.

Per ulteriori informazioni sulla creazione di una regola con S3 RTC, consulta Abilitazione di S3 Replication Time Control (S3 RTC).

Parametri di replica con S3 RTC

Le regole di replica per le quali è abilitato S3 Replication Time Control (S3 RTC) pubblicano i parametri di replica. Con i parametri di replica, puoi monitorare il numero totale di operazioni API S3 in attesa di replica, la dimensione totale degli oggetti in attesa di replica e il tempo massimo di replica

nella regione di destinazione e il numero totale di operazioni che non sono state replicate. Quindi puoi monitorare separatamente ogni set di dati replicato.

I parametri di replica sono disponibili entro 15 minuti dall'attivazione di S3 RTC. I parametri di replica sono disponibili tramite la console Amazon S3, l'API Amazon S3, AWS gli SDK, () e Amazon.AWS Command Line InterfaceAWS CLI CloudWatch Per ulteriori informazioni, consulta Monitoraggio delle metriche con Amazon CloudWatch.

Per ulteriori informazioni sulla ricerca dei parametri di replica tramite la console di Amazon S3, consulta Visualizzazione delle metriche di replica utilizzando la console Amazon S3.

Utilizzo delle notifiche di eventi Amazon S3 per tenere traccia degli oggetti di replica

Puoi tenere traccia del tempo di replica per gli oggetti che non sono stati replicati entro 15 minuti monitorando specifiche notifiche di eventi pubblicate da S3 Replication Time Control (S3 RTC). Questi eventi vengono pubblicati quando un oggetto idoneo per la replica mediante S3 RTC non viene replicato entro 15 minuti e quando l'oggetto viene quindi replicato una volta superata la soglia di 15 minuti.

Gli eventi di replica sono disponibili entro 15 minuti dall'abilitazione di S3 RTC. Gli eventi Amazon S3 sono disponibili tramite Amazon SQS, Amazon SNS o. AWS Lambda Per ulteriori informazioni, consulta Notifiche di eventi Amazon S3.

Best practice e linee guida per S3 RTC

Per la replica dei dati in Amazon S3 usando S3 Replication Time Control (S3 RTC), attieniti alle seguenti linee guida di best practice per ottimizzare le prestazioni di replica dei carichi di lavoro.

Argomenti

- Linee guida sulle prestazioni per la frequenza di richieste e la replica di Amazon S3
- Stima delle frequenze di richieste di replica
- Superamento dei limiti di velocità di trasferimento dati S3 RTC
- AWS KMS tassi di richiesta di replica di oggetti crittografati

Linee guida sulle prestazioni per la frequenza di richieste e la replica di Amazon S3

Le applicazioni possono raggiungere migliaia di transazioni al secondo nelle prestazioni di richiesta durante il caricamento e il recupero di storage da Amazon S3. Ad esempio, un'applicazione può raggiungere almeno 3.500 richieste PUT/COPY/POST/DELETE o 5.500 richieste GET/HEAD al

secondo per prefisso in un bucket S3, incluse le richieste che la replica S3 effettua per tuo conto. Non esistono limiti al numero di prefissi in un bucket. È possibile aumentare le proprie performance in lettura o scrittura parallelizzando le scritture. Ad esempio, se crei 10 prefissi in un bucket S3 per parallelizzare le letture, è possibile dimensionare le prestazioni di lettura a 55.000 richieste di lettura al secondo.

Amazon S3 esegue il dimensionamento automatico in risposta a frequenze di richieste sostenute oltre queste linee guida o a frequenze di richieste sostenute in contemporanea a richieste LIST. Mentre Amazon S3 si ottimizza internamente per la nuova frequenza di richieste, potresti ricevere temporaneamente risposte HTTP 503 fino al completamento dell'ottimizzazione. Questo potrebbe verificarsi con gli aumenti delle frequenze di richiesta al secondo o quando si abilita per la prima volta S3 RTC. Durante questi periodi, la latenza di replica potrebbe aumentare. Il contratto sul livello di servizio (SLA) di S3 RTC non si applica ai periodi di tempo in cui vengono superate le linee guida sulle prestazioni di Amazon S3 per le richieste al secondo.

Il contratto sul livello di servizio S3 RTC non si applica neanche ai periodi di tempo in cui la velocità di trasferimento dati di replica supera il limite predefinito di 1 Gbps. Se prevedi che la velocità di trasferimento della replica superi 1 Gbps, puoi contattare il Centro AWS Support o utilizzare Service Quotas per richiedere un aumento del limite.

Stima delle frequenze di richieste di replica

La frequenza di richieste totale, incluse le richieste effettuate dalla replica Amazon S3 per tuo conto, deve rientrare nelle linee guida sulla frequenza di richieste Amazon S3 per i bucket di origine e di destinazione della replica. Per ogni oggetto replicato, la replica di Amazon S3 esegue fino a cinque richieste GET/HEAD e una richiesta PUT al bucket di origine e una richiesta PUT al bucket di destinazione.

Ad esempio, se prevedi di replicare 100 oggetti al secondo, la replica di Amazon S3 potrebbe eseguire ulteriori 100 richieste PUT per tuo conto per un totale di 200 PUT al secondo nel bucket S3 di origine. La replica Amazon S3 potrebbe anche eseguire fino a 500 richieste GET/HEAD (5 richieste GET/HEAD per ogni oggetto replicato).

> **Note**
>
> Vengono addebitati i costi per una sola richiesta PUT per oggetto replicato. Per ulteriori informazioni, consulta le informazioni sui prezzi nelle domande frequenti di Amazon S3 sulla replica.

Superamento dei limiti di velocità di trasferimento dati S3 RTC

Se prevedi che la velocità di trasferimento dati di S3 Replication Time Control superi il limite predefinito di 1 Gbps, contatta il Centro AWS Support o utilizza Service Quotas per richiedere un aumento del limite.

AWS KMS tassi di richiesta di replica di oggetti crittografati

Quando si replicano oggetti crittografati con crittografia lato server (SSE-KMS) utilizzando la replica AWS Key Management Service Amazon S3, si applicano limiti di richieste al secondo ().AWS KMS AWS KMS potrebbe rifiutare una richiesta altrimenti valida perché la frequenza delle richieste supera il limite per il numero di richieste al secondo. Quando una richiesta viene limitata, AWS KMS restituisce un errore. `ThrottlingException` Il limite di frequenza delle AWS KMS richieste si applica alle richieste effettuate direttamente e alle richieste effettuate dalla replica di Amazon S3 per tuo conto.

Ad esempio, se prevedi di replicare 1.000 oggetti al secondo, puoi sottrarre 2.000 richieste dal limite di frequenza delle richieste. AWS KMS La frequenza di richieste al secondo risultante è disponibile per i AWS KMS carichi di lavoro, esclusa la replica. Puoi utilizzare i parametri di AWS KMS richiesta in Amazon CloudWatch per monitorare il tasso totale di AWS KMS richieste sul tuo Account AWS.

Abilitazione di S3 Replication Time Control (S3 RTC)

Amazon S3 Replication Time Control (S3 RTC) permette di soddisfare i requisiti aziendali o di conformità per la replica dei dati fornendo visibilità sui tempi di replica di Amazon S3. S3 RTC replica la maggior parte degli oggetti caricati in Amazon S3 in pochi secondi e il 99,99% di tali oggetti entro 15 minuti.

Con S3 RTC, puoi monitorare il numero totale e la dimensione totale degli oggetti in attesa di replica e il tempo massimo di replica nella regione di destinazione. Le metriche di replica sono disponibili tramite AWS Management ConsoleAmazon CloudWatch User Guide. Per ulteriori informazioni, consulta the section called “Metriche di replica S3 in CloudWatch ”.

Utilizzo della console S3

Per step-by-step istruzioni, consulta. Configurazione della replica per i bucket di origine e di destinazione di proprietà dello stesso account Questo argomento fornisce istruzioni per abilitare S3 RTC nella configurazione di replica quando i bucket sono di proprietà uguale o diversa. Account AWS

Usare il AWS CLI

Per utilizzare la AWS CLI replica di oggetti con S3 RTC abilitato, devi creare bucket, abilitare il controllo delle versioni sui bucket, creare un ruolo IAM che autorizzi Amazon S3 a replicare oggetti e aggiungere la configurazione di replica al bucket di origine. È necessario che S3 Replication Time Control (S3 RTC) sia abilitato nella configurazione di replica.

Come replicare con S3 RTC abilitato (AWS CLI)

- Nell'esempio seguente sono impostati `ReplicationTime` e `Metric` e viene aggiunta la configurazione di replica al bucket di origine.

```
{
    "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` e `ReplicationTime:Time:Minutes` possono avere solo 15 come un valore valido.

Utilizzo dell' AWS SDK for Java

Di seguito è riportato un esempio Java per aggiungere la configurazione di replica con S3 Replication Time Control (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);
  }
}
```

Per ulteriori informazioni, consulta Rispetto dei requisiti di conformità utilizzando S3 Replication Time Control (S3 RTC).

Replica di oggetti crittografati (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS)

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva negli SDK and. AWS Command Line Interface AWS Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

Quando si replicano oggetti che sono stati crittografati utilizzando la crittografia lato server, è necessario prestare particolare attenzione. Amazon S3 supporta i seguenti tipi di crittografia lato server:

- Crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3)
- Crittografia lato AWS Key Management Service server AWS KMS con chiavi () (SSE-KMS)
- Crittografia lato server a doppio livello con chiavi (DSSE-KMS) AWS KMS
- Crittografia lato server con chiavi fornite dal cliente (SSE-C)

Per ulteriori informazioni sulla crittografia lato server, consulta the section called “Crittografia lato server”.

Questo argomento spiega le autorizzazioni necessarie per indirizzare Amazon S3 a replicare oggetti che sono stati crittografati utilizzando la crittografia lato server. Questo argomento fornisce anche

elementi di configurazione aggiuntivi che è possibile aggiungere ed esempi di policy AWS Identity and Access Management (IAM) che concedono le autorizzazioni necessarie per la replica di oggetti crittografati.

Per un esempio con step-by-step istruzioni, vedere. Abilitazione della replica per oggetti crittografati Per informazioni sulla creazione di una configurazione di replica, consulta Panoramica sulla replica degli oggetti.

> **Note**
>
> Puoi usare più regioni AWS KMS keys in Amazon S3. Tuttavia, Amazon S3 attualmente tratta le chiavi multi-regione come se fossero chiavi a regione singola e non utilizza le caratteristiche multi-regione della chiave. Per ulteriori informazioni, consulta Utilizzo delle chiavi multi-regione nella Guida per gli sviluppatori di AWS Key Management Service .

Argomenti

- In che modo la crittografia predefinita del bucket influisce sulla replica
- Replica di oggetti crittografati con SSE-C
- Replica di oggetti crittografati con SSE-S3, SSE-KMS o DSSE-KMS
- Abilitazione della replica per oggetti crittografati

In che modo la crittografia predefinita del bucket influisce sulla replica

Una volta abilitata la crittografia predefinita per un bucket di destinazione della replica, si applica il seguente comportamento di crittografia:

- Se gli oggetti nel bucket di origine non sono crittografati, gli oggetti replicati nel bucket di destinazione vengono crittografati in base alle impostazioni di crittografia predefinita del bucket di destinazione. Di conseguenza, i tag di entità (ETag) degli oggetti di origine differiscono dagli ETag degli oggetti di replica. Se disponi di applicazioni che utilizzano ETag, devi aggiornarle per tenere conto di questa differenza.
- Se gli oggetti nel bucket di origine sono crittografati utilizzando la crittografia lato server con chiavi gestite Amazon S3 (SSE-S3), la crittografia lato server con chiavi () (SSE-KMS AWS KMS) o la crittografia lato server a doppio livello con AWS Key Management Service AWS KMS chiavi (DSSE-KMS), gli oggetti di replica nel bucket di destinazione utilizzano lo stesso tipo di crittografia

degli oggetti di origine. Le impostazioni della crittografia predefinita del bucket di destinazione non vengono utilizzate.

Replica di oggetti crittografati con SSE-C

Utilizzando la crittografia lato server con chiavi fornite dal cliente (SSE-C), è possibile gestire le proprie chiavi di crittografia proprietarie. Con SSE-C, puoi gestire le chiavi mentre Amazon S3 si occupa del processo di crittografia e decrittografia. È necessario fornire una chiave di crittografia come parte della richiesta, ma non è necessario scrivere codice per eseguire la crittografia o la decrittografia degli oggetti. Quando carichi un oggetto, Amazon S3 ne esegue la crittografia utilizzando la chiave che hai specificato. Quindi Amazon S3 elimina la chiave dalla memoria. Quando viene recuperato un oggetto, è necessario fornire la stessa chiave di crittografia come parte della richiesta. Per ulteriori informazioni, consulta the section called “Crittografia lato server con chiavi fornite dal cliente (SSE-C)”.

S3 Replication supporta oggetti crittografati con SSE-C. Puoi configurare la replica di oggetti SSE-C nella console Amazon S3 o con gli AWS SDK, allo stesso modo in cui configuri la replica per oggetti non crittografati. Non sono disponibili autorizzazioni SSE-C aggiuntive oltre a quelle attualmente richieste per la replica.

La replica S3 replica automaticamente gli oggetti crittografati con SSE-C appena caricati, se idonei, secondo la configurazione di replica S3 specificata. Per la replica di oggetti esistenti nei bucket, utilizza la replica in batch in S3. Per ulteriori informazioni sulla replica di oggetti, consulta the section called “Configurazione della replica in tempo reale” e the section called “Replica di oggetti esistenti”.

Non sono previsti costi aggiuntivi per la replica di oggetti SSE-C. Per informazioni dettagliate sui prezzi della replica, consulta la pagina Prezzi di Amazon S3.

Replica di oggetti crittografati con SSE-S3, SSE-KMS o DSSE-KMS

Per impostazione predefinita, Amazon S3 non replica gli oggetti crittografati con SSE-KMS o DSSE-KMS. Questa sezione illustra un'ulteriore configurazione che puoi aggiungere per fare in modo che Amazon S3 replichi questi oggetti.

Per un esempio con istruzioni, consulta. step-by-step Abilitazione della replica per oggetti crittografati Per informazioni sulla creazione di una configurazione di replica, consulta Panoramica sulla replica degli oggetti.

Specifica di informazioni aggiuntive nella configurazione di replica

Nella configurazione di replica, è necessario eseguire queste operazioni:

- Nell'`Destination`elemento della configurazione di replica, aggiungi l'ID della chiave simmetrica gestita dal AWS KMS cliente che desideri che Amazon S3 utilizzi per crittografare le repliche degli oggetti, come mostrato nel seguente esempio di configurazione di replica.
- Specifica esplicitamente la funzione abilitando la replica di oggetti crittografati mediante le chiavi KMS (SSE-KMS o DSSE-KMS). Per attivare, aggiungi l'elemento `SourceSelectionCriteria`, come mostrato nel seguente esempio di configurazione della replica.

```
<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
 Regione AWS as the destination bucket.</ReplicaKmsKeyID>
          </EncryptionConfiguration>
       </Destination>
      ...
   </Rule>
</ReplicationConfiguration>
```

> **Important**
>
> La chiave KMS deve essere stata creata nello stesso modo in cui sono stati creati i bucket di destinazione. Regione AWS
> Chiave KMS deve essere valida. L'operazione `PutBucketReplication` dell'API non controlla la validità delle chiavi KMS. Se utilizzi una chiave KMS non valida, viene restituito il codice di stato HTTP `200 OK` in risposta, ma la replica non riesce.

Nell'esempio seguente viene illustrata una configurazione di replica che include gli elementi di configurazione opzionali. Questa configurazione di replica ha una regola. La regola si applica agli oggetti con il prefisso della chiave `Tax`. Amazon S3 utilizza l'ID AWS KMS key specificato per crittografare queste repliche di oggetti.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration>
   <Role>arn:aws:iam::account-id:role/role-name</Role>
   <Rule>
      <ID>Rule-1</ID>
      <Priority>1</Priority>
      <Status>Enabled</Status>
      <DeleteMarkerReplication>
         <Status>Disabled</Status>
      </DeleteMarkerReplication>
      <Filter>
         <Prefix>Tax</Prefix>
      </Filter>
      <Destination>
         <Bucket>arn:aws:s3:::example-s3-destination-bucket</Bucket>
         <EncryptionConfiguration>
            <ReplicaKmsKeyID>AWS KMS key ARN or Key Alias ARN that's in the
 same Regione 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>
```

Concessione di autorizzazioni aggiuntive per il ruolo IAM

Per replicare oggetti crittografati a riposo utilizzando SSE-S3, SSE-KMS o DSSE-KMS, concedi le seguenti autorizzazioni aggiuntive al ruolo (IAM) specificato nella configurazione di replica. AWS Identity and Access Management Queste autorizzazioni vengono concesse aggiornando la policy di autorizzazione associata al ruolo IAM.

- Operazione **`s3:GetObjectVersionForReplication`** per gli oggetti di origine: consente ad Amazon S3 di replicare gli oggetti non crittografati e gli oggetti creati con la crittografia lato server mediante le chiavi SSE-S3, SSE-KMS o DSSE-KMS.

  > **Note**
  >
  > Ti consigliamo di utilizzare l'operazione `s3:GetObjectVersionForReplication` anziché l'operazione `s3:GetObjectVersion` in quanto `s3:GetObjectVersionForReplication` concede ad Amazon S3 solo le autorizzazioni minime necessarie per la replica. Inoltre, l'operazione `s3:GetObjectVersion` permette la replica di oggetti non crittografati e crittografati SSE-S3, ma non di oggetti crittografati utilizzando le chiavi KMS (SSE-KMS o DSSE-KMS).

- **`kms:Decryptkms:Encrypt`** AWS KMS e azioni per le chiavi KMS
  - È necessario concedere le autorizzazioni `kms:Decrypt` per la AWS KMS key utilizzata per decrittografare l'oggetto di origine.
  - È necessario concedere le autorizzazioni `kms:Encrypt` per la AWS KMS key utilizzata per crittografare la replica dell'oggetto.
- Operazione **`kms:GenerateDataKey`** per la replica di oggetti in testo normale: se stai replicando oggetti di testo normale in un bucket con la crittografia SSE-KMS o DSSE-KMS abilitata per impostazione predefinita, devi includere l'autorizzazione `kms:GenerateDataKey` per il contesto di crittografia di destinazione e la chiave KMS nella policy IAM.

Ti consigliamo di limitare queste autorizzazioni solo ai bucket e agli oggetti di destinazione utilizzando AWS KMS le chiavi di condizione. Il Account AWS titolare del ruolo IAM deve disporre delle `kms:Decrypt` autorizzazioni `kms:Encrypt` e delle azioni per le chiavi KMS elencate nella policy. Se le chiavi KMS sono di proprietà di un altro Account AWS, il proprietario delle chiavi KMS deve concedere queste autorizzazioni al proprietario del Account AWS ruolo IAM. Per ulteriori informazioni sulla gestione dell'accesso a queste chiavi KMS, consulta Using IAM Policies with AWS KMS nella Developer Guide. AWS Key Management Service

Chiavi bucket S3 e replica

Per utilizzare la replica con una chiave S3 Bucket, la AWS KMS key policy per la chiave KMS utilizzata per crittografare la replica dell'oggetto deve includere l'autorizzazione per il principale chiamante. `kms:Decrypt` La chiamata a `kms:Decrypt` verifica l'integrità della chiave bucket S3

prima del suo utilizzo. Per ulteriori informazioni, consulta Utilizzo di una chiave bucket S3 con la replica.

Quando una chiave del bucket S3 è abilitata per il bucket di origine o di destinazione, il contesto di crittografia sarà il nome della risorsa Amazon (ARN) del bucket e non l'ARN dell'oggetto, ad esempio `arn:aws:s3:::bucket_ARN`. Dovrai aggiornare le policy IAM per utilizzare l'ARN del bucket per il contesto di crittografia:

```
"kms:EncryptionContext:aws:s3:arn": [
"arn:aws:s3:::bucket_ARN"
]
```

Per ulteriori informazioni, consulta Contesto di crittografia (`x-amz-server-side-encryption-context`) (nella sezione relativa a REST API) e Modifiche alla nota prima dell'abilitazione di una chiave bucket S3.

Policy di esempio: utilizzo di SSE-S3 e SSE-KMS con la replica

Le policy IAM di esempio riportate di seguito mostrano le istruzioni per utilizzare SSE-S3 e SSE-KMS con la replica.

Example - Utilizzo di SSE-KMS con bucket di destinazione separati

La seguente policy di esempio mostra le istruzioni per utilizzare SSE-KMS con bucket di destinazione separati.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action": ["kms:Decrypt"],
         "Effect": "Allow",
         "Condition": {
             "StringLike": {
                  "kms:ViaService": "s3.source-bucket-region.amazonaws.com",
                  "kms:EncryptionContext:aws:s3:arn": [
                      "arn:aws:s3:::example-s3-source-bucket/key-prefix1*"
                  ]
             }
          },
          "Resource": [
             "List of AWS KMS key ARNs that are used to encrypt source objects."
```

```
            ]
        },
        {
            "Action": ["kms:Encrypt"],
            "Effect": "Allow",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": "s3.destination-bucket-1-region.amazonaws.com",
                    "kms:EncryptionContext:aws:s3:arn": [
                        "arn:aws:s3:::example-s3-destination-bucket1/key-prefix1*"
                    ]
            }
            },
            "Resource": [
                "AWS KMS key ARNs (in the same Regione AWS as destination bucket 1). Used
 to encrypt object replicas created in destination bucket 1."
            ]
        },
        {
            "Action": ["kms:Encrypt"],
            "Effect": "Allow",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": "s3.destination-bucket-2-region.amazonaws.com",
                    "kms:EncryptionContext:aws:s3:arn": [
                        "arn:aws:s3:::example-s3-destination-bucket2/key-prefix1*"
                    ]
            }
            },
            "Resource": [
                "AWS KMS key ARNs (in the same Regione AWS as destination bucket 2). Used
 to encrypt object replicas created in destination bucket 2."
            ]
        }
    ]
}
```

Example - Replica di oggetti creati con SSE-S3 e SSE-KMS

Di seguito è riportata una policy IAM completa che concede le autorizzazioni necessarie per la replica di oggetti non crittografati, oggetti creati con SSE-S3 e oggetti creati con SSE-KMS.

```
{
```

```
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::example-s3-source-bucket"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl"
         ],
         "Resource":[
            "arn:aws:s3:::example-s3-source-bucket/key-prefix1*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete"
         ],
         "Resource":"arn:aws:s3:::example-s3-destination-bucket/key-prefix1*"
      },
      {
         "Action":[
            "kms:Decrypt"
         ],
         "Effect":"Allow",
         "Condition":{
            "StringLike":{
               "kms:ViaService":"s3.source-bucket-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn":[
                  "arn:aws:s3:::example-s3-source-bucket/key-prefix1*"
               ]
            }
         },
         "Resource":[
```

```
            "List of the AWS KMS key ARNs that are used to encrypt source objects."
         ]
      },
      {
         "Action":[
            "kms:Encrypt"
         ],
         "Effect":"Allow",
         "Condition":{
            "StringLike":{
               "kms:ViaService":"s3.destination-bucket-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn":[
                  "arn:aws:s3:::example-s3-destination-bucket/prefix1*"
               ]
            }
         },
         "Resource":[
            "AWS KMS key ARNs (in the same Regione AWS as the destination bucket) to
 use for encrypting object replicas"
         ]
      }
   ]
}
```

Example - Replica oggetti con chiavi bucket S3

Di seguito è riportata una policy IAM completa che concede le autorizzazioni necessarie per la replica degli oggetti con chiavi bucket S3.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::example-s3-source-bucket"
         ]
      },
      {
```

```
        "Effect":"Allow",
        "Action":[
           "s3:GetObjectVersionForReplication",
           "s3:GetObjectVersionAcl"
        ],
        "Resource":[
           "arn:aws:s3:::example-s3-source-bucket/key-prefix1*"
        ]
     },
     {
        "Effect":"Allow",
        "Action":[
           "s3:ReplicateObject",
           "s3:ReplicateDelete"
        ],
        "Resource":"arn:aws:s3:::example-s3-destination-bucket/key-prefix1*"
     },
     {
        "Action":[
           "kms:Decrypt"
        ],
        "Effect":"Allow",
        "Condition":{
           "StringLike":{
              "kms:ViaService":"s3.source-bucket-region.amazonaws.com",
              "kms:EncryptionContext:aws:s3:arn":[
                 "arn:aws:s3:::example-s3-source-bucket"
              ]
           }
        },
        "Resource":[
          "List of the AWS KMS key ARNs that are used to encrypt source objects."
        ]
     },
     {
        "Action":[
           "kms:Encrypt"
        ],
        "Effect":"Allow",
        "Condition":{
           "StringLike":{
              "kms:ViaService":"s3.destination-bucket-region.amazonaws.com",
              "kms:EncryptionContext:aws:s3:arn":[
                 "arn:aws:s3:::example-s3-destination-bucket"
```

```
                ]
            }
        },
        "Resource":[
            "AWS KMS key ARNs (in the same Regione AWS as the destination bucket) to
 use for encrypting object replicas"
        ]
      }
   ]
}
```

Concessione di autorizzazioni aggiuntive per scenari multi-account

In uno scenario con più account, in cui i bucket di origine e di destinazione sono di proprietà di diversi Account AWS, puoi utilizzare una chiave KMS per crittografare le repliche degli oggetti. Tuttavia, il proprietario della chiave KMS deve concedere al proprietario del bucket di origine l'autorizzazione per utilizzare la chiave KMS.

> **Note**
>
> Se è necessario replicare i dati SSE-KMS su più account, la regola di replica deve specificare una chiave gestita dal cliente per l'account di destinazione. AWS KMS Chiavi gestite da AWSnon consentono l'utilizzo tra account e pertanto non possono essere utilizzati per eseguire la replica tra account.

Per concedere al proprietario del bucket di origine l'autorizzazione per l'utilizzo della chiave KMS (console AWS KMS)

1. Accedi AWS Management Console e apri la AWS KMS console all'indirizzo https://console.aws.amazon.com/kms.
2. Per modificare la Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Per visualizzare le chiavi nell'account creato e gestito dall'utente, nel riquadro di navigazione, seleziona Chiavi gestite dal cliente.
4. Scegli la chiave KMS.
5. In Configurazione generale, seleziona la scheda Policy delle chiavi.
6. Scorri verso il basso fino a Altro. Account AWS

7. Scegli Aggiungi altro Account AWS.

   Viene visualizzata la Account AWS finestra di dialogo Altro.

8. Nella finestra di dialogo, scegli Aggiungi un altro Account AWS. Per arn:aws:iam::, inserisci l'ID account del bucket di origine.

9. Seleziona Salvataggio delle modifiche.

Per concedere al proprietario del bucket di origine l'autorizzazione per l'utilizzo della chiave KMS (AWS CLI)

- Per informazioni sul comando `put-key-policy` AWS Command Line Interface (AWS CLI), vedete put-key-policynella Guida ai AWS CLI comandi. Per ulteriori informazioni sull'operazione dell'API sottostante, consulta `PutKeyPolicy` in PutKeyPolicy API ReferenceAWS Key Management Service (Guida di riferimento per l'API di).

AWS KMS considerazioni sulle quote di transazione

Quando si aggiungono molti nuovi oggetti con AWS KMS crittografia dopo aver abilitato la replica tra regioni (CRR), è possibile che si verifichi una limitazione (errori HTTP). `503 Service Unavailable` La limitazione (della larghezza di banda della rete) si verifica quando il numero di transazioni AWS KMS al secondo supera la quota corrente. Per ulteriori informazioni, consulta Quote nella Guida per gli sviluppatori di AWS Key Management Service .

Per richiedere un aumento della quota, è possibile utilizzare Service Quotas. Per ulteriori informazioni, consulta la sezione Richiesta di un aumento di quota. Se Service Quotas non è supportato nella tua regione, apri una AWS Support richiesta.

Abilitazione della replica per oggetti crittografati

Per impostazione predefinita, Amazon S3 non replica oggetti crittografati utilizzando la crittografia lato server con AWS Key Management Service () chiavi (SSE-KMS AWS KMS) o la crittografia lato server a due livelli con chiavi (DSSE-KMS). AWS KMS Per replicare gli oggetti crittografati con SSE-KMS o DSS-KMS, devi modificare la configurazione di replica del bucket per indicare ad Amazon S3 di replicare questi oggetti. Questo esempio spiega come utilizzare la console Amazon S3 e AWS Command Line Interface (AWS CLI) per modificare la configurazione della replica dei bucket per consentire la replica di oggetti crittografati.

Per ulteriori informazioni, consulta Replica di oggetti crittografati (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

> **Note**
>
> Quando una chiave del bucket S3 è abilitata per il bucket di origine o di destinazione, il contesto di crittografia sarà il nome della risorsa Amazon (ARN) del bucket e non l'ARN dell'oggetto. Dovrai aggiornare le policy IAM per utilizzare l'ARN del bucket per il contesto di crittografia. Per ulteriori informazioni, consulta Chiavi bucket S3 e replica.

> **Note**
>
> Puoi usare più regioni AWS KMS keys in Amazon S3. Tuttavia, Amazon S3 attualmente tratta le chiavi multi-regione come se fossero chiavi a regione singola e non utilizza le caratteristiche multi-regione della chiave. Per ulteriori informazioni, consulta Utilizzo delle chiavi multi-regione nella Guida per gli sviluppatori di AWS Key Management Service .

Utilizzo della console S3

Per step-by-step istruzioni, consulta. Configurazione della replica per i bucket di origine e di destinazione di proprietà dello stesso account Questo argomento fornisce istruzioni per impostare una configurazione di replica quando i bucket sono di proprietà uguale o diversa. Account AWS

Usare il AWS CLI

Per replicare oggetti crittografati con AWS CLI, effettuate le seguenti operazioni:

- Crea i bucket di origine e di destinazione e abilita il controllo delle versioni per questi bucket.
- Crea un ruolo di servizio AWS Identity and Access Management (IAM) che autorizzi Amazon S3 a replicare oggetti. Le autorizzazioni del ruolo IAM includono le autorizzazioni necessarie per replicare gli oggetti crittografati.
- Aggiungi una configurazione di replica al bucket di origine. La configurazione di replica fornisce informazioni relative alla replica di oggetti crittografati con le chiavi KMS.
- Aggiungi gli oggetti crittografati al bucket di origine.
- Esegui il test della configurazione per verificare che gli oggetti crittografati vengano replicati nel bucket di destinazione.

Le procedure seguenti ti guidano attraverso questo processo.

Per replicare gli oggetti crittografati lato server (AWS CLI)

1. In questo esempio crei entrambi i bucket *example-s3-source-bucket* e *example-s3-destination-bucket* nello stesso Account AWS. Imposti anche un profilo di credenziali per la AWS CLI. In questo esempio si utilizza il nome del profilo *acctA*.

   Per ulteriori informazioni sull'impostazione dei profili di credenziali, consulta Named Profiles nella Guida per l' AWS Command Line Interface utente. Per usare i comandi in questo esempio, sostituisci *user input placeholders* con le tue informazioni.

2. Usa i seguenti comandi per creare il bucket *DOC-EXAMPLE-SOURCE-BUCKET* e abilitare il controllo delle versioni. Il seguente comando di esempio crea il bucket *DOC-EXAMPLE-SOURCE-BUCKET* nella regione Stati Uniti orientali (Virginia settentrionale) (`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. Usa i seguenti comandi per creare il bucket *DOC-EXAMPLE-DESTINATION-BUCKET* e abilitare il controllo delle versioni. Il seguente comando di esempio crea il bucket *DOC-EXAMPLE-DESTINATION-BUCKET* nella regione Stati Uniti occidentali (Oregon) (`us-west-2`).

   > **Note**
   >
   > Per impostare una configurazione di replica quando entrambi *DOC-EXAMPLE-SOURCE-BUCKET*i *DOC-EXAMPLE-DESTINATION-BUCKET*bucket sono nello stesso Account AWS, si utilizza lo stesso profilo. In questo esempio viene utilizzato *acctA*. Per configurare la replica quando i bucket sono di proprietà di Account AWS diversi, occorre specificare profili differenti per ciascuno di essi.

```
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. Quindi, crea un ruolo di servizio IAM. Specifica questo ruolo nella configurazione di replica che aggiungi al bucket *DOC-EXAMPLE-SOURCE-BUCKET* in un secondo momento. Amazon S3 assume questo ruolo per replicare gli oggetti per tuo conto. Il ruolo IAM si crea in due fasi:

   - Creazione di un ruolo del servizio
   - Collegare una policy di autorizzazione al ruolo.

   a. Per creare un ruolo di servizio IAM, procedi come segue:

      i. Copiare la seguente policy di attendibilità e salvala in un file denominato `s3-role-trust-policy-kmsobj.json` nella directory corrente sul computer locale. Questa policy fornisce le autorizzazioni ai principali del servizio Amazon S3 per assumere il ruolo in modo che Amazon S3 possa eseguire attività per tuo conto.

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

      ii. Usa il comando seguente per creare il ruolo:

```
$ aws iam create-role \
--role-name replicationRolekmsobj \
--assume-role-policy-document file://s3-role-trust-policy-kmsobj.json  \
--profile acctA
```

b. Quindi, collega una policy di autorizzazione al ruolo. Questa policy di accesso concede le autorizzazioni per varie operazioni su oggetti e bucket Amazon S3.

i. Copiare la seguente policy di autorizzazioni e salvarla in un file denominato `s3-role-permissions-policykmsobj.json` nella directory corrente sul computer locale. Crea un ruolo IAM e successivamente collegalo alla policy.

> **Important**
>
> Nella politica di autorizzazione, si specificano gli ID delle AWS KMS chiavi che verranno utilizzati per la crittografia dei *example-s3-source-bucket* bucket and. *example-s3-destination-bucket È necessario creare due chiavi KMS separate per i bucket example-s3-source-bucket example-s3-destination-bucket.* AWS KMS keys non sono condivise al di fuori di quella in cui sono state create. Regione AWS

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "s3:ListBucket",
            "s3:GetReplicationConfiguration",
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::example-s3-source-bucket",
            "arn:aws:s3:::example-s3-source-bucket/*"
         ]
      },
      {
```

```
        "Action":[
           "s3:ReplicateObject",
           "s3:ReplicateDelete",
           "s3:ReplicateTags"
        ],
        "Effect":"Allow",
        "Condition":{
           "StringLikeIfExists":{
              "s3:x-amz-server-side-encryption":[
                 "aws:kms",
                 "AES256",
                 "aws:kms:dsse"
              ],
              "s3:x-amz-server-side-encryption-aws-kms-key-id":[
                 "AWS KMS key IDs(in ARN format) to use for encrypting
object replicas"
              ]
           }
        },
        "Resource":"arn:aws:s3:::example-s3-destination-bucket/*"
     },
     {
        "Action":[
           "kms:Decrypt"
        ],
        "Effect":"Allow",
        "Condition":{
           "StringLike":{
              "kms:ViaService":"s3.us-east-1.amazonaws.com",
              "kms:EncryptionContext:aws:s3:arn":[
                 "arn:aws:s3:::example-s3-source-bucket/*"
              ]
           }
        },
        "Resource":[
           "AWS KMS key IDs(in ARN format) used to encrypt source
objects."
        ]
     },
     {
        "Action":[
           "kms:Encrypt"
        ],
        "Effect":"Allow",
```

```
         "Condition":{
            "StringLike":{
               "kms:ViaService":"s3.us-west-2.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn":[
                  "arn:aws:s3:::example-s3-destination-bucket/*"
               ]
            }
         },
         "Resource":[
            "AWS KMS key IDs(in ARN format) to use for encrypting object
 replicas"
         ]
      }
   ]
}
```

ii. Creare una policy e collegarla al ruolo.

```
$ aws iam put-role-policy \
--role-name replicationRolekmsobj \
--policy-document file://s3-role-permissions-policykmsobj.json \
--policy-name replicationRolechangeownerPolicy \
--profile acctA
```

5. Quindi, aggiungi la seguente configurazione di replica al bucket *example-s3-source-bucket* che indica ad Amazon S3 di replicare gli oggetti con prefisso `Tax/` nel bucket *example-s3-destination-bucket*.

> **Important**
>
> Nella configurazione di replica dovrai specificare il ruolo IAM che Amazon S3 può assumere. Puoi effettuare questa operazione solo se disponi dell'autorizzazione `iam:PassRole`. Il profilo specificato nel comando della CLI deve disporre di questa autorizzazione. Per ulteriori informazioni, consulta Concessione di autorizzazioni utente per il passaggio di un ruolo a un Servizio AWS nella Guida per l'utente di IAM.

```
<ReplicationConfiguration>
 <Role>IAM-Role-ARN</Role>
 <Rule>
   <Priority>1</Priority>
```

```
    <DeleteMarkerReplication>
       <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter>
       <Prefix>Tax</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <SourceSelectionCriteria>
      <SseKmsEncryptedObjects>
        <Status>Enabled</Status>
      </SseKmsEncryptedObjects>
    </SourceSelectionCriteria>
    <Destination>
      <Bucket>arn:aws:s3:::example-s3-destination-bucket</Bucket>
      <EncryptionConfiguration>
        <ReplicaKmsKeyID>AWS KMS key IDs to use for encrypting object replicas</
ReplicaKmsKeyID>
      </EncryptionConfiguration>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Per aggiungere una configurazione di replica al bucket *example-s3-source-bucket*, procedi come segue:

a. AWS CLI Richiede di specificare la configurazione di replica come JSON. Salvare il seguente JSON in un file (`replication.json`) nella directory corrente sul computer locale.

```
{
   "Role":"IAM-Role-ARN",
   "Rules":[
      {
         "Status":"Enabled",
         "Priority":1,
         "DeleteMarkerReplication":{
            "Status":"Disabled"
         },
         "Filter":{
            "Prefix":"Tax"
         },
         "Destination":{
```

```
            "Bucket":"arn:aws:s3:::example-s3-destination-bucket",
            "EncryptionConfiguration":{
               "ReplicaKmsKeyID":"AWS KMS key IDs (in ARN format) to use for
 encrypting object replicas"
            }
         },
         "SourceSelectionCriteria":{
            "SseKmsEncryptedObjects":{
               "Status":"Enabled"
            }
         }
      }
   ]
}
```

b. Modifica il JSON per fornire valori per il bucket *example-s3-destination-bucket*, *AWS KMS key IDs (in ARN format)* e *IAM-role-ARN*. Salvare le modifiche.

c. Esegui il comando seguente per aggiungere la configurazione di replica al bucket *example-s3-source-bucket*. Assicurati di fornire il nome del bucket di *example-s3-source-bucket*.

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket example-s3-source-bucket \
--profile acctA
```

6. Esegui il test della configurazione per verificare che gli oggetti crittografati vengano replicati. Nella console di Amazon S3 effettuare quanto segue:

a. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

b. Nel bucket di *example-s3-source-bucket*, crea una cartella denominata `Tax`.

c. Aggiungere oggetti campione alla cartella. Assicurati di scegliere l'opzione di crittografia e specifica la chiave KMS per crittografare gli oggetti.

d. Verifica che il bucket *example-s3-destination-bucket* contenga le repliche dell'oggetto e che queste vengano crittografate utilizzando la chiave KMS specificata nella configurazione. Per ulteriori informazioni, consulta the section called “Ottenimento dello stato della replica”.

Utilizzo degli SDK AWS

Per un esempio di codice che illustra come aggiungere una configurazione di replica, consulta Utilizzo degli SDK AWS. La configurazione della replica deve essere modificata di conseguenza.

Per informazioni concettuali, consulta Replica di oggetti crittografati (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

Replica delle modifiche ai metadati con la sincronizzazione delle modifiche alla replica Amazon S3

La sincronizzazione delle modifiche alle repliche di Amazon S3 può aiutarti a mantenere replicati i metadati degli oggetti come tag, ACL e le impostazioni di blocco oggetti tra repliche e oggetti di origine. Per impostazione predefinita, Amazon S3 replica i metadati dagli oggetti di origine solo alle repliche. Quando la sincronizzazione delle modifiche alla replica è abilitata, Amazon S3 replica le modifiche apportate ai metadati apportate alle copie di replica nell'oggetto di origine, rendendo la replica bidirezionale.

Abilitazione della sincronizzazione delle modifiche alla replica

Puoi utilizzare la sincronizzazione delle modifiche alla replica Amazon S3 con regole di replica nuove o esistenti. Puoi applicare tale replica a un intero bucket S3 o agli oggetti Amazon S3 che hanno un prefisso specifico.

Per abilitare la sincronizzazione delle modifiche alla replica utilizzando la console di Amazon S3, consulta Esempi di configurazione della replica in tempo reale. Questo argomento fornisce istruzioni per abilitare la sincronizzazione delle modifiche alla replica nella configurazione di replica quando i bucket sono di proprietà uguale o diversa. Account AWS

Per abilitare la sincronizzazione delle modifiche alla replica utilizzando AWS Command Line Interface (AWS CLI), è necessario aggiungere una configurazione di replica al bucket contenente le repliche con enabled. `ReplicaModifications` *Per configurare la replica bidirezionale, create una regola di replica dal bucket di origine (example-s3-bucket1) al bucket contenente le repliche (example-s3-bucket2). Quindi, crea una seconda regola di replica dal bucket contenente le repliche (example-s3-bucket2) al bucket di origine (example-s3-bucket1).* I bucket Regioni AWS possono essere uguali o diversi.

> **Note**
>
> Per replicare le modifiche ai metadati di replica, quali liste di controllo degli accessi (ACL) degli oggetti, tag oggetto o impostazioni di Blocco oggetto sugli oggetti replicati devi abilitare

la sincronizzazione delle modifiche alle repliche su entrambi i bucket. Come tutte le regole di replica, queste regole possono essere applicate all'intero bucket Amazon S3 o a un sottoinsieme di oggetti Amazon S3 filtrati per prefisso o tag oggetto.

Nella seguente configurazione di esempio, Amazon S3 replica le modifiche ai metadati con il prefisso *Tax* nel bucket *example-s3-bucket*, che conterrebbe gli oggetti di origine.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "SourceSelectionCriteria": {
                "ReplicaModifications":{
                    "Status": "Enabled"
                }
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

Per istruzioni complete sulla creazione di regole di replica utilizzando il, consulta. AWS CLIConfigurazione della replica per i bucket di origine e di destinazione di proprietà dello stesso account

Replica dei contrassegni di eliminazione tra i bucket

Per impostazione predefinita, quando la replica S3 è abilitata e un oggetto viene eliminato nel bucket di origine, Amazon S3 aggiunge un contrassegno di eliminazione solo nel bucket di origine. Ciò permette di proteggere i dati da eliminazioni indesiderate.

Se hai abilitato la replica dei contrassegni di eliminazione, questi contrassegni saranno copiati nei bucket di destinazione e Amazon S3 si comporterà come se l'oggetto fosse stato eliminato sia

nei bucket di origine che in quelli di destinazione. Per ulteriori informazioni sul funzionamento dei contrassegni di eliminazione, consulta Utilizzo dei contrassegni di eliminazione.

> **Note**
>
> La replica dei contrassegni di eliminazione non è supportata per le regole di replica basate su tag. La replica dei contrassegni di eliminazione non è compatibile con il contratto di servizio di 15 minuti concesso quando si utilizza S3 Replication Time Control.

Se non utilizzi la versione più recente della configurazione di replica, le operazioni di eliminazione influiranno sulla replica in modo diverso. Per ulteriori informazioni, consulta Effetto delle operazioni di eliminazione sulla replica.

Abilitazione della replica dei contrassegni di eliminazione

Puoi iniziare a utilizzare la replica dei contrassegni di eliminazione con una regola di replica nuova o esistente. Puoi applicare tale replica a un intero bucket S3 o agli oggetti Amazon S3 che hanno un prefisso specifico.

> **Note**
>
> Quando abiliti la replica dei marker di eliminazione e il tuo bucket ha una regola di scadenza del ciclo di vita S3, i marker di eliminazione aggiunti dalla regola di scadenza di S3 Lifecycle non verranno replicati nel bucket di destinazione.

Per abilitare la replica dei contrassegni di eliminazione utilizzando la console di Amazon S3, consulta Utilizzo della console S3. Questo argomento fornisce istruzioni per abilitare la replica dei marker di eliminazione nella configurazione di replica quando i bucket sono di proprietà uguale o diversa. Account AWS

Per abilitare la replica dei marker di eliminazione utilizzando AWS Command Line Interface (AWS CLI), è necessario aggiungere una configurazione di replica al bucket di origine con enabled. `DeleteMarkerReplication`

Nella configurazione di esempio seguente, i contrassegni di eliminazione vengono replicati nel bucket di destinazione *`DOC-EXAMPLE-BUCKET`* per gli oggetti sotto il prefisso *`Imposta`*.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "DeleteMarkerReplication": {
                "Status": "Enabled"
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

Per istruzioni complete sulla creazione di regole di replica tramite AWS CLI, Configurazione della replica per i bucket di origine e di destinazione di proprietà dello stesso account consultate la sezione Procedure dettagliate sulla replica.

Gestire o sospendere la replica in tempo reale

La replica in tempo reale è la copia automatica e asincrona di oggetti tra bucket uguali o diversi. Regioni AWS Dopo aver impostato la configurazione di replica, Amazon S3 replica gli oggetti appena creati e gli aggiornamenti degli oggetti da un bucket di origine a uno o più bucket di destinazione specificati.

Per aggiungere regole di replica al bucket di origine, viene utilizzata la console di Amazon S3. Le regole di replica definiscono gli oggetti del bucket di origine da replicare e i bucket o i bucket di destinazione in cui vengono archiviati gli oggetti replicati. Per ulteriori informazioni sulla replica, consulta Panoramica sulla replica degli oggetti.

È possibile gestire le regole di replica nella pagina Replica. Puoi aggiungere, visualizzare, abilitare, disabilitare o eliminare le regole di replica. È inoltre possibile modificare la priorità delle regole di replica. Per informazioni sull'aggiunta di regole di replica a un bucket, consulta Utilizzo della console S3.

Per gestire le regole di replica per un bucket S3 utilizzando la console Amazon S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nella scheda General Purpose bucket, scegli il nome del bucket che desideri.
4. Scegli la scheda Gestione, quindi scorri verso il basso fino a Regole di replica.
5. È possibile modificare le regole di replica nei seguenti modi:
   - Per abilitare o disabilitare una regola di replica, scegli il pulsante di opzione a sinistra della regola. Nel menu Azioni, scegli Abilita regola o Disabilita regola. Puoi anche disabilitare, abilitare o eliminare tutte le regole nel bucket dal menu Azioni.

     > **Note**
     >
     > Se si disabilita una regola di replica e successivamente la si riattiva, tutti gli oggetti nuovi o modificati che non sono stati replicati mentre la regola era disabilitata non vengono replicati automaticamente quando la regola viene riattivata. Per replicare questi oggetti, è necessario utilizzare S3 Batch Replication. Per ulteriori informazioni, consulta the section called “Replica di oggetti esistenti”.

   - Per modificare la priorità di una regola, scegli il pulsante di opzione a sinistra della regola, quindi scegli Modifica regola.

     È necessario impostare le priorità delle regole per evitare i conflitti causati dagli oggetti inclusi nell'ambito di più regole. In caso di regole sovrapposte, Amazon S3 utilizza la priorità delle regole per determinare quale regola applicare. Più elevato è il numero, maggiore è la priorità. Per ulteriori informazioni sulla priorità delle regole, consulta Configurazione di replica.

Sospensione o arresto della replica

Per sospendere temporaneamente la replica e farla riprendere automaticamente in un secondo momento, è possibile utilizzare l'azione in. `aws:s3:bucket-pause-replication` AWS Fault Injection ServicePer ulteriori informazioni, consulta aws:s3:bucket-pause-replicatione metti in pausa la replica S3 nella Guida per l'utente.AWS Fault Injection Service

Per interrompere la replica in Amazon S3, consigliamo di disabilitare le regole di replica. Se disabiliti una regola di replica e successivamente riattivi la regola, tutti gli oggetti nuovi o modificati che non sono stati replicati mentre la regola era disabilitata non vengono replicati automaticamente quando la regola viene riattivata. Per replicare questi oggetti, è necessario utilizzare S3 Batch Replication. Per ulteriori informazioni, consulta the section called “Replica di oggetti esistenti”.

La replica si interromperà anche se rimuovi il ruolo AWS Identity and Access Management (IAM), le autorizzazioni AWS Key Management Service (AWS KMS) o le autorizzazioni della bucket policy che concedono ad Amazon S3 le autorizzazioni richieste. Tuttavia, non consigliamo questi approcci perché impediscono la replica. Amazon S3 segnala lo stato di replica per gli oggetti interessati come `FAILED`. Se le autorizzazioni vengono successivamente ripristinate, gli oggetti contrassegnati come non *`FAILED`* vengono replicati automaticamente. Per replicare questi oggetti, è necessario utilizzare S3 Batch Replication.

Monitoraggio dell'avanzamento con le metriche di replica e le notifiche eventi di Amazon S3

I parametri di replica S3 forniscono parametri dettagliati per le regole nella configurazione di replica. Con le metriche di replica, è possibile monitorare l' minute-by-minute avanzamento tenendo traccia dei byte in sospeso, delle operazioni in sospeso, delle operazioni che non hanno avuto esito positivo e della latenza di replica.

I parametri di replica S3 vengono attivati automaticamente quando si abilita il controllo del tempo di replica di S3 (S3 RTC). Puoi anche abilitare le metriche di Replica Amazon S3 indipendentemente da S3 RTC durante la creazione o la modifica di una regola. La funzionalità di controllo del tempo di replica di S3 (S3 RTC) include altre funzionalità, ad esempio un Accordo sul livello di servizio (SLA) e notifiche per soglie non raggiunte. Per ulteriori informazioni, consulta Rispetto dei requisiti di conformità utilizzando S3 Replication Time Control (S3 RTC).

Le metriche relative ai byte in attesa di replica, alle operazioni attesa di replica e alla latenza di replica si applicano solo ai nuovi oggetti replicati con la replica tra Regioni di S3 (S3 CRR) o la replica nella stessa Regione di S3 (S3 SRR). La metrica delle operazioni con replica non riuscita tiene traccia sia dei nuovi oggetti replicati con S3 CRR o S3 SRR sia degli oggetti esistenti replicati con Replica Amazon S3. A supporto delle procedure di risoluzione dei problemi di configurazione, puoi anche configurare la funzionalità Notifiche eventi Amazon S3 per ricevere eventi relativi agli errori di replica.

Se abilitati, i parametri di replica di S3 pubblicano i seguenti parametri su Amazon: CloudWatch

- Byte in attesa di replica: il numero totale di byte di oggetti in attesa di replica per una determinata regola di replica.
- Latenza di replica: il numero massimo di secondi entro i quali i bucket di destinazione della replica sono in ritardo rispetto al bucket di origine per una determinata regola di replica.
- Operazioni in attesa di replica: il numero di operazioni in attesa di replica per una determinata regola di replica. Questa metrica tiene traccia delle operazioni relative a oggetti, contrassegni di eliminazione, tag, liste di controllo degli accessi (ACL) e blocco degli oggetti S3.
- Operazioni di replica non riuscite: il numero di operazioni di replica non riuscite per una determinata regola di replica. Questa metrica tiene traccia delle operazioni relative a oggetti, contrassegni di eliminazione, tag, ACL e blocco degli oggetti. A differenza delle altre metriche di replica, questa metrica si applica sia ai nuovi oggetti replicati con S3 CRR o S3 SRR sia agli oggetti esistenti replicati con Replica Amazon S3.

> **Note**
>
> Operazioni di replica non riuscite tiene traccia degli errori di Replica Amazon S3 aggregati a intervalli di un minuto. Per identificare gli oggetti specifici la cui replica non è riuscita e i relativi motivi, iscriviti all'evento `OperationFailedReplication` mediante la funzionalità Notifiche eventi Amazon S3. Per ulteriori informazioni, consulta Ricezione di eventi di errore di replica con notifiche di eventi Amazon S3.

Se un processo non viene eseguito affatto, le metriche non vengono inviate ad Amazon CloudWatch. Ad esempio, il processo non verrà eseguito se non disponi delle autorizzazioni necessarie per eseguire un processo Replica Amazon S3 o se i tag o il prefisso nella configurazione della replica non corrispondono.

Argomenti

- Abilitazione dei parametri di replica S3
- Ricezione di eventi di errore di replica con notifiche di eventi Amazon S3
- Visualizzazione dei parametri di replica con S3 Storage Lens
- Visualizzazione delle metriche di replica utilizzando la console Amazon S3
- Motivi degli errori di replica Amazon S3
- Ottenimento delle informazioni sullo stato della replica

Abilitazione dei parametri di replica S3

Puoi iniziare a utilizzare i parametri di replica S3 con una regola di replica nuova o esistente. Puoi decidere di applicare la regola di replica a un intero bucket S3 o a oggetti Amazon S3 con un prefisso o un tag specifico.

In questo argomento vengono fornite le istruzioni per abilitare le metriche di Replica S3 nella configurazione della replica quando i bucket di origine e destinazione sono di proprietà dello stesso o di diversi Account AWS.

Per abilitare i parametri di replica utilizzando il comando AWS Command Line Interface (AWS CLI), devi aggiungere una configurazione di replica al bucket di origine con enabled. `Metrics` In questa configurazione di esempio, gli oggetti con il prefisso *`Tax`* vengono replicati nel bucket di destinazione *`DOC-EXAMPLE-BUCKET`* e vengono generate le metriche per tali oggetti.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "Metrics": {
                    "Status": "Enabled"
                }
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

Per istruzioni complete sulla creazione delle regole di replica, consulta Configurazione della replica per i bucket di origine e di destinazione di proprietà dello stesso account.

Per ulteriori informazioni sulla visualizzazione dei parametri di replica nella console S3, consulta Visualizzazione delle metriche di replica utilizzando la console Amazon S3.

> **Note**
>
> I parametri di replica S3 vengono fatturati alla stessa tariffa dei parametri personalizzati di Amazon. CloudWatch Per ulteriori informazioni, consulta i CloudWatchprezzi di Amazon.

Ricezione di eventi di errore di replica con notifiche di eventi Amazon S3

La funzionalità Notifiche eventi Amazon S3 può inviare un avviso nei casi in cui gli oggetti non vengono replicati nella Regione AWS di destinazione. Gli eventi Amazon S3 sono disponibili tramite Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS) oppure. AWS Lambda Per ulteriori informazioni, consulta the section called “Notifiche di eventi Amazon S3”.

Per l'elenco dei codici di errore acquisiti dalla funzionalità Notifiche eventi Amazon S3, consulta Motivi degli errori di replica Amazon S3.

Visualizzazione dei parametri di replica con S3 Storage Lens

Per ottenere parametri dettagliati per Replica S3, inclusi i parametri relativi al conteggio delle regole di replica, puoi utilizzare Amazon S3 Storage Lens. S3 Storage Lens è una funzionalità di analisi dell'archiviazione su cloud che puoi utilizzare per avere una panoramica completa a livello di organizzazione sull'utilizzo e sulle attività relative all'archiviazione di oggetti. Per ulteriori informazioni, consulta Utilizzo di S3 Storage Lens per proteggere i dati. Per un elenco completo delle metriche, consulta il glossario delle metriche di S3 Storage Lens.

Visualizzazione delle metriche di replica utilizzando la console Amazon S3

Esistono tre tipi di CloudWatch parametri Amazon per Amazon S3: parametri di storage, parametri di richiesta e parametri di replica. I parametri di replica S3 vengono attivati automaticamente quando si abilita la replica con S3 Replication Time Control (S3 RTC) utilizzando l'API o Amazon S3. AWS Management Console Puoi anche abilitare le metriche di Replica Amazon S3 indipendentemente da S3 RTC durante la creazione o la modifica di una regola.

I parametri di replica tengono traccia degli ID regola della configurazione di replica. Un ID regola di replica può essere specifico per un prefisso, un tag o una combinazione di entrambi.

Per ulteriori informazioni sui CloudWatch parametri per Amazon S3, consulta. Monitoraggio delle metriche con Amazon CloudWatch

Prerequisiti

Abilita una regola di replica contenente le metriche di Replica Amazon S3.

Per visualizzare i parametri di replica

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket). Nell'elenco Bucket seleziona il nome del bucket contenente gli oggetti per cui si desidera ottenere le metriche di replica.
3. Seleziona la scheda Parametri.
4. In Parametri di replica, seleziona Regole di replica.
5. Seleziona Visualizza grafici.

   Amazon S3 visualizza i grafici relativi a latenza di replica (in secondi), byte in attesa di replica, operazioni in attesa di replica e operazioni di replica non riuscite.

A questo punto puoi visualizzare le metriche relative alla replica per latenza di replica (in secondi), operazioni in attesa di replica, byte in attesa di replica e operazioni di replica non riuscite per le regole selezionate. Se utilizzi S3 Replication Time Control, Amazon CloudWatch inizia a riportare i parametri di replica 15 minuti dopo aver abilitato S3 RTC sulla rispettiva regola di replica. Puoi visualizzare i parametri di replica sulla console Amazon S3 o sulla console. CloudWatch Per ulteriori informazioni, consulta Parametri di replica con S3 RTC.

> **Note**
>
> Puoi anche visualizzare i parametri dettagliati per la replica S3 nella console Amazon S3 utilizzando Amazon S3 Storage Lens. S3 Storage Lens è una funzionalità di analisi dell'archiviazione su cloud che puoi utilizzare per avere una panoramica completa a livello di organizzazione sull'utilizzo e sulle attività relative all'archiviazione di oggetti. Per ulteriori informazioni, consulta Utilizzo di S3 Storage Lens per proteggere i dati. Per un elenco completo dei parametri, consulta il glossario dei parametri di S3 Storage Lens.

Motivi degli errori di replica Amazon S3

La seguente tabella elenca i motivi degli errori di replica in Amazon S3. Puoi visualizzare questi motivi ricevendo l'evento FailureReason con la funzionalità Notifiche eventi Amazon S3. Puoi

ricevere notifiche di eventi S3 tramite Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS) oppure. AWS Lambda Per ulteriori informazioni, consulta Notifiche di eventi Amazon S3.

Puoi visualizzare questi motivi di errore anche nei report di completamento della replica in batch in S3. Per ulteriori informazioni, consulta Report di completamento della replica in batch.

Motivo dell'errore di replica	Descrizione
`AssumeRoleNotPermitted`	Amazon S3 non può assumere il ruolo AWS Identity and Access Management (IAM) specificato nella configurazione di replica o nel job Batch Operations.
`DstBucketInvalidRegion`	Il bucket di destinazione non è Regione AWS uguale a quello specificato dal job Batch Operations. Questo errore è specifico per la replica in batch.
`DstBucketNotFound`	Amazon S3 non è in grado di trovare il bucket di destinazione specificato nella configura zione della replica.
`DstBucketObjectLockConfigMissing`	Per replicare gli oggetti da un bucket di origine con la funzionalità di blocco degli oggetti abilitata, anche la destinazione deve avere il blocco degli oggetti abilitato. Questo errore indica che il blocco degli oggetti potrebbe non essere abilitato nel bucket di destinazione. Per ulteriori informazioni, consulta Considerazioni su Object Lock.
`DstBucketUnversioned`	Il controllo delle versioni non è abilitato per il bucket di destinazione S3. Per replicare gli oggetti con la funzionalità Replica Amazon S3, abilita il controllo delle versioni per bucket di destinazione.

Motivo dell'errore di replica	Descrizione
`DstDelObjNotPermitted`	Amazon S3 non è in grado di replicare i contrassegni di eliminazione nel bucket di destinazione. È possibile che manchi l'autorizzazione `s3:ReplicateDelete` per il bucket di destinazione.
`DstKmsKeyInvalidState`	La chiave AWS Key Management Service (AWS KMS) per il bucket di destinazione non è in uno stato valido. Rivedi e abilita la AWS KMS chiave richiesta. Per ulteriori informazioni sulla gestione delle AWS KMS chiavi, consulta Key states of AWS KMS keys nella AWS Key Management Service Developer Guide.
`DstKmsKeyNotFound`	La AWS KMS chiave configurata per il bucket di destinazione nella configurazione di replica non esiste.
`DstMultipartCompleteNotPermitted`	Amazon S3 non è in grado di completare i caricamenti in più parti degli oggetti nel bucket di destinazione. È possibile che manchi l'autorizzazione `s3:ReplicateObject` per il bucket di destinazione.
`DstMultipartInitNotPermitted`	Amazon S3 non è in grado di avviare i caricamenti in più parti degli oggetti nel bucket di destinazione. È possibile che manchi l'autorizzazione `s3:ReplicateObject` per il bucket di destinazione.
`DstMultipartPartUploadNotPermitted`	Amazon S3 non è in grado di caricare oggetti in più parti nel bucket di destinazione. È possibile che manchi l'autorizzazione `s3:ReplicateObject` per il bucket di destinazione.

Motivo dell'errore di replica	Descrizione
`DstObjectHardDeleted`	S3 Batch Replication non supporta la ripetizione della replica di oggetti eliminati con l'ID versione dell'oggetto del bucket di destinazione. Questo errore è specifico per la replica in batch.
`DstPutAclNotPermitted`	Amazon S3 non è in grado di replicare le liste di controllo degli accessi (ACL) dell'oggetto nel bucket di destinazione. È possibile che manchi l'autorizzazione `s3:ReplicateObject` per il bucket di destinazione.
`DstPutLegalHoldNotPermitted`	Amazon S3 non è in grado di impostare un blocco degli oggetti a fini legali per l'oggetto di destinazione durante la replica di oggetti immutabili. È possibile che manchi l'autorizzazione `s3:PutObjectLegalHold` per il bucket di destinazione. Per ulteriori informazioni, consulta Blocchi a fini giudiziari.
`DstPutObjectNotPermitted`	Amazon S3 non è in grado di replicare oggetti nel bucket di destinazione. È possibile che manchino le autorizzazioni `s3:ReplicateObject` o `s3:ObjectOwnerOverrideToBucketOwner` per il bucket di destinazione.
`DstPutTaggingNotPermitted`	Amazon S3 non è in grado di replicare tag di oggetti nel bucket di destinazione. È possibile che manchi l'autorizzazione `s3:ReplicateObject` per il bucket di destinazione.
`DstVersionNotFound`	Amazon S3 non è in grado di trovare la versione dell'oggetto richiesta nel bucket di destinazione per cui devono essere replicati i metadati.

Motivo dell'errore di replica	Descrizione
`InitiateReplicationNotPermitted`	Amazon S3 non è in grado di avviare la replica sugli oggetti. È possibile che manchi l'autorizzazione `s3:InitiateReplication` per il processo Operazioni in batch. Questo errore è specifico per la replica in batch.
`SrcBucketInvalidRegion`	Il bucket di origine non è Regione AWS uguale a quello specificato dal job Batch Operations. Questo errore è specifico per la replica in batch.
`SrcBucketNotFound`	Amazon S3 non è in grado di trovare il bucket di origine.
`SrcBucketReplicationConfigMissing`	Amazon S3 non è riuscito a trovare una configurazione della replica per il bucket di origine.

Motivo dell'errore di replica	Descrizione
`SrcGetAclNotPermitted`	Amazon S3 non è in grado di accedere all'oggetto nel bucket di origine per la replica. È possibile che manchi l'autorizzazione `s3:GetObjectVersionAcl` per l'oggetto del bucket di origine. Gli oggetti nel bucket di origine devono essere di proprietà del proprietario del bucket. Se gli ACL sono abilitati, verifica se Proprietà dell'oggetto è impostata su Proprietario del bucket preferito o Autore dell'oggetto. Se Proprietà dell'oggetto è impostata su Proprietario del bucket preferito , gli oggetti del bucket di origine devono avere l'ACL `bucket-owner-full-control` affinché il proprietario del bucket diventi il proprietario dell'oggetto. L'account di origine può acquisire la proprietà di tutti gli oggetti nel relativo bucket impostando Proprietà dell'oggetto su Proprietario del bucket preferito e disabilitando gli ACL.
`SrcGetLegalHoldNotPermitted`	Amazon S3 non è in grado di accedere alle informazioni di conservazione legale di S3 Object Lock.
`SrcGetObjectNotPermitted`	Amazon S3 non è in grado di accedere all'oggetto nel bucket di origine per la replica. È possibile che manchi l'autorizzazione `s3:GetObjectVersionForRepli cation` per il bucket di origine.
`SrcGetRetentionNotPermitted`	Amazon S3 non è in grado di accedere alle informazioni del periodo di conservazione di S3 Object Lock.

Motivo dell'errore di replica	Descrizione
`SrcGetTaggingNotPermitted`	Amazon S3 non è in grado di accedere alle informazioni sui tag di oggetto dal bucket di origine. È possibile che manchi l'autorizzazione `s3:GetObjectVersionTagging` per il bucket di origine.
`SrcHeadObjectNotPermitted`	Amazon S3 non è in grado di recuperare i metadati dell'oggetto dal bucket di origine. È possibile che manchi l'autorizzazione `s3:GetObjectVersionForReplication` per il bucket di origine.
`SrcKeyNotFound`	Amazon S3 non è in grado di trovare la chiave dell'oggetto di origine da replicare. L'oggetto di origine potrebbe essere stato eliminato prima del completamento della replica.
`SrcKmsKeyInvalidState`	La AWS KMS chiave per il bucket di origine non è in uno stato valido. Rivedi e abilita la AWS KMS chiave richiesta. Per ulteriori informazioni sulla gestione delle AWS KMS chiavi, consulta Key states of AWS KMS keys nella AWS Key Management Service Developer Guide.
`SrcObjectNotEligible`	Alcuni oggetti non sono idonei per la replica. Ciò può essere dovuto alla classe di archiviazione dell'oggetto o ai tag dell'oggetto che non corrispondono alla configurazione di replica.
`SrcObjectNotFound`	L'oggetto di origine non esiste.
`SrcReplicationNotPending`	Amazon S3 ha già replicato questo oggetto. Questo oggetto non è più in attesa di replica.

Motivo dell'errore di replica	Descrizione
`SrcVersionNotFound`	Amazon S3 non è in grado di trovare la versione dell'oggetto di origine da replicare. La versione dell'oggetto di origine potrebbe essere stato eliminato prima del completam ento della replica.

Argomenti correlati

Impostazione delle autorizzazioni per la replica in tempo reale

Risoluzione dei problemi nella replica

Ottenimento delle informazioni sullo stato della replica

Lo stato della replica consente di determinare lo stato corrente di un oggetto da replicare. Lo stato della replica di un oggetto di origine restituirà `PENDING`, `COMPLETED` o `FAILED`. Lo stato della replica di una replica restituirà `REPLICA`.

Argomenti

- Panoramica dello stato della replica
- Stato della replica in caso di replica su più bucket di destinazione
- Stato della replica se è abilitata la sincronizzazione della modifica alla replica Amazon S3
- Ricerca dello stato di replica

Panoramica dello stato della replica

Nella replica, esistono un bucket di origine in cui si configura la replica e un bucket di destinazione in cui Amazon S3 replica gli oggetti. Quando richiedi un oggetto (tramite l'oggetto `GET`) o i metadati di un oggetto (tramite l'oggetto `HEAD`) da questi bucket, Amazon S3 restituisce l'intestazione `x-amz-replication-status` nella risposta:

- Quando richiedi un oggetto dal bucket di origine, Amazon S3 restituisce l'intestazione `x-amz-replication-status` se l'oggetto nella richiesta è idoneo per la replica.

  Supponi, ad esempio, che nella configurazione di replica venga specificato il prefisso di oggetto `TaxDocs` che indica ad Amazon S3 di replicare solo gli oggetti con il prefisso del nome della chiave

`TaxDocs`. Tutti gli oggetti caricati che hanno questo prefisso del nome della chiave, ad esempio `TaxDocs/document1.pdf`, verranno replicati. Per le richieste di oggetti con questo prefisso del nome della chiave, Amazon S3 restituisce l'intestazione `x-amz-replication-status` con uno dei valori seguenti per lo stato della replica dell'oggetto: `PENDING`, `COMPLETED` o `FAILED`.

> **Note**
>
> Se la replica dell'oggetto ha esito negativo dopo il caricamento di un oggetto, non è possibile provare a eseguirla di nuovo. È necessario caricare di nuovo l'oggetto. Gli oggetti passano a uno stato `FAILED` per problemi dovuti ad esempio alla mancanza di autorizzazioni per il ruolo di replica, autorizzazioni AWS KMS o autorizzazioni di bucket. In caso di errori temporanei, ad esempio se un bucket o una regione non è disponibile, lo stato della replica non passerà a `FAILED`, ma rimarrà `PENDING`. Dopo che la risorsa è tornata online, S3 riprenderà la replica di tali oggetti.

- Quando richiedi un oggetto dal bucket di destinazione, se l'oggetto nella richiesta è una replica creata da Amazon S3, Amazon S3 restituisce l'intestazione `x-amz-replication-status` con il valore `REPLICA`.

> **Note**
>
> Prima di eliminare un oggetto da un bucket di origine in cui è abilitata la replica, è consigliabile controllare lo stato della replica per assicurarsi che l'oggetto sia stato replicato. Se nel bucket di origine è abilitata la configurazione del ciclo di vita, Amazon S3 sospende tutte le operazioni del ciclo di vita fino a quando non contrassegna lo stato degli oggetti come `COMPLETED` o `FAILED`.

Stato della replica in caso di replica su più bucket di destinazione

Quando si replicano oggetti in più bucket di destinazione, l'intestazione `x-amz-replication-status` funziona in modo diverso. L'intestazione dell'oggetto di origine restituisce un valore `COMPLETED` solo se la replica ha esito positivo su tutte le destinazioni. L'intestazione rimane al valore `PENDING` fino al completamento della replica per tutte le destinazioni. Se la replica su una o più destinazioni non riesce, viene restituita l'intestazione `FAILED`.

Stato della replica se è abilitata la sincronizzazione della modifica alla replica Amazon S3

Quando le regole di replica abilitano la sincronizzazione delle modifiche alla replica Amazon S3, le repliche possono riportare stati diversi da `REPLICA`. Se le modifiche dei metadati sono in corso di replica, l'intestazione `x-amz-replication-status` restituisce `PENDING`. Se la sincronizzazione delle modifiche della replica non riesce a replicare i metadati, l'intestazione restituisce `FAILED`. Se i metadati vengono replicati correttamente, le repliche restituiscono l'intestazione `REPLICA`.

Ricerca dello stato di replica

Per visualizzare lo stato della replica degli oggetti in un bucket, è possibile utilizzare lo strumento Inventario Amazon S3. Amazon S3 invia un file CSV al bucket di destinazione specificato nella configurazione dell'inventario. Puoi anche utilizzare Amazon Athena per eseguire una query sullo stato della replica nel report di inventario. Per ulteriori informazioni su Inventario Amazon S3, consulta Amazon S3 Inventory.

Puoi anche trovare lo stato di replica degli oggetti utilizzando la console, AWS Command Line Interface (AWS CLI) o l' AWS SDK.

Utilizzo della console S3

Nella console S3 puoi visualizzare lo stato della replica di un oggetto nella pagina Dettagli dell'oggetto in Panoramica della gestione degli oggetti.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket, seleziona il nome del bucket.
3. Nell'elenco Oggetti, seleziona il nome dell'oggetto.
4. Nella scheda Properties (Proprietà) cerca Object management overview (Panoramica della gestione degli oggetti) dove puoi vedere lo stato della replica.

Utilizzando il AWS CLI

Per richiamare i metadati dell'oggetto, utilizza il comando `head-object` come riportato di seguito.

```
aws s3api head-object --bucket source-bucket --key object-key --version-id object-version-id
```

Il comando restituisce i metadati dell'oggetto, incluso l'elemento `ReplicationStatus` come illustrato nella risposta di esempio seguente.

```
{
   "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":{

   }
}
```

Utilizzo degli AWS SDK

I seguenti frammenti di codice ottengono lo stato di replica rispettivamente con AWS SDK for Java e AWS SDK for .NET.

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);
```

Replica di oggetti esistenti con S3 Batch Replication

Utilizzando S3 Batch Replication, è possibile replicare i seguenti tipi di oggetti:

- Oggetti che esistevano prima dell'implementazione di una configurazione di replica
- Oggetti che sono stati replicati in precedenza
- Oggetti la cui replica non è riuscita

È possibile replicare questi oggetti su richiesta utilizzando un processo Batch Operations. La replica in batch di S3 è diversa dalla replica live, che replica in modo continuo e automatico nuovi oggetti tra i bucket Amazon S3.

Per iniziare a usare Batch Replication, puoi:

- Avvia la replica in batch per una nuova regola o destinazione di replica: puoi creare un processo di replica in batch una tantum quando crei la prima regola in una nuova configurazione di replica o quando aggiungi una nuova destinazione a una configurazione esistente tramite la console Amazon S3.
- Avvia la replica in batch per una configurazione di replica esistente: puoi creare un nuovo processo di replica in batch utilizzando S3 Batch Operations tramite la console Amazon S3, il AWS Command Line Interface (AWS CLI), gli SDK AWS o l'API REST di Amazon S3.

Al termine del processo Batch Replication, viene visualizzato un report di completamento. Per ulteriori informazioni su come utilizzare il report per esaminare il processo, consulta la sezione Monitoraggio dei rapporti sullo stato e sul completamento dei processi.

Considerazioni su S3 Batch Replication

- Il bucket di origine deve avere una configurazione di replica esistente. Per abilitare la replica, consulta le sezioni Configurazione della replica in tempo reale e Esempi di configurazione della replica in tempo reale.
- Se hai configurato S3 Lifecycle per il tuo bucket, ti consigliamo di disabilitare le regole del ciclo di vita mentre il job Batch Replication è attivo. In questo modo è possibile garantire la parità tra i bucket di origine e di destinazione. In caso contrario, questi bucket potrebbero divergere e il bucket di destinazione non sarà una replica esatta del bucket di origine. Si consideri ad esempio lo scenario riportato di seguito:

  - Il bucket di origine contiene più versioni di un oggetto e un marker di eliminazione su quell'oggetto.
  - I bucket di origine e destinazione dispongono di una configurazione del ciclo di vita per rimuovere i contrassegni di eliminazione scaduti.

  In questo scenario, Batch Replication potrebbe replicare il marker di eliminazione nel bucket di destinazione prima di replicare le versioni dell'oggetto. Questo comportamento potrebbe far sì che la configurazione del ciclo di vita contrassegni il marker di eliminazione come scaduto e che il marker di eliminazione venga rimosso dal bucket di destinazione prima che le versioni dell'oggetto vengano replicate.

- Il ruolo AWS Identity and Access Management (IAM) specificato per eseguire il processo Batch Operations deve disporre delle autorizzazioni necessarie per eseguire l'operazione di replica batch sottostante. Per ulteriori informazioni sulla creazione dei ruoli IAM, consulta la sezione Configurazione delle policy IAM per Batch Replication.
- La replica in batch richiede un manifesto, che può essere generato da Amazon S3. Il manifesto generato deve essere archiviato nello stesso Regione AWS bucket di origine. Se scegli di non generare il manifesto, puoi fornire un report di Amazon S3 Inventory o un file CSV contenente gli oggetti che desideri replicare.
- La replica in batch non supporta la replica di oggetti che sono stati eliminati con l'ID di versione dell'oggetto dal bucket di destinazione. Per replicare nuovamente questi oggetti è possibile copiare gli oggetti di origine presenti con un processo di copia in batch. La copia di tali oggetti sul posto crea nuove versioni degli oggetti nel bucket di origine e avvia automaticamente la replica nel bucket di destinazione. L'eliminazione e la ricreazione del bucket di destinazione non avviano la replica.

  Per ulteriori informazioni su Batch Copy, vedereEsempi che utilizzano operazioni in batch per copiare oggetti.
- Se utilizzi una regola di replica sul bucket S3, assicurati di aggiornare la configurazione di replica concedendo al ruolo IAM associato alla regola di replica le autorizzazioni appropriate per replicare gli oggetti. Questo ruolo IAM deve disporre delle autorizzazioni necessarie per eseguire la replica sia sul bucket di origine che su quello di destinazione.
- Se invii più processi di replica in batch per lo stesso bucket in un breve lasso di tempo, Amazon S3 eseguirà tali processi contemporaneamente.
- Se invii più lavori di replica in batch per due bucket diversi, tieni presente che Amazon S3 potrebbe non eseguire tutti i job contemporaneamente. Se superi il numero di processi di replica in batch che possono essere eseguiti contemporaneamente sul tuo account, Amazon S3 metterà in pausa

i processi con priorità più bassa per lavorare su quelli con priorità più alta. Una volta completati gli elementi con priorità più alta, tutti i lavori sospesi torneranno attivi.

- La replica in batch non è supportata per gli oggetti archiviati nelle classi di storage S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive.
- Per replicare in batch gli oggetti S3 Intelligent-Tiering archiviati nei livelli di storage Archive Access o Deep Archive Access, devi prima avviare una richiesta di ripristino e attendere che gli oggetti vengano spostati sul livello Frequent Access.

Specifica di un manifesto per un processo Batch Replication

Un manifesto è un oggetto Amazon S3 contenente le chiavi degli oggetti su cui Amazon S3 deve agire. Se desideri creare un processo di replica in batch, devi fornire un manifesto generato dall'utente o fare in modo che Amazon S3 generi un manifesto in base alla tua configurazione di replica.

Se fornisci un manifesto generato dall'utente, deve avere la forma di un report di inventario Amazon S3 o di un file CSV. Se gli oggetti nel manifesto sono in un bucket con versione, è necessario specificare gli ID versione per gli oggetti. Verrà replicato solo l'oggetto con l'ID di versione specificato nel manifesto. Per ulteriori informazioni sulla specifica di un manifesto, consulta la sezione Specifica di un manifest.

Se scegli di fare in modo che Amazon S3 generi un file manifesto per tuo conto, gli oggetti elencati utilizzeranno lo stesso bucket di origine, lo stesso prefisso e gli stessi tag di tutte le configurazioni di replica del bucket di origine. Con un manifesto generato, Amazon S3 replicherà tutte le versioni idonee dei tuoi oggetti.

> **Note**
>
> Se scegli di fare in modo che Amazon S3 generi il manifesto, il manifesto deve essere archiviato nello Regione AWS stesso bucket di origine.

Filtri per i processi Batch Replication

Quando si crea un processo di replica in batch, è possibile specificare facoltativamente filtri aggiuntivi, come la data di creazione dell'oggetto e lo stato della replica, per ridurre l'ambito del lavoro.

Puoi filtrare gli oggetti da replicare in base al valore `ObjectReplicationStatuses` fornendo uno o più dei seguenti valori:

- `"NONE"`: indica che Amazon S3 non ha mai tentato di replicare l'oggetto in precedenza.
- `"FAILED"`— Indica che Amazon S3 ha già tentato, ma non è riuscito, di replicare l'oggetto in precedenza.
- `"COMPLETED"`: indica che Amazon S3 ha replicato correttamente l'oggetto in precedenza.
- `"REPLICA"`— Indica che si tratta di un oggetto di replica che Amazon S3 ha replicato da un'altra origine.

Per ulteriori informazioni sugli stati di replica, consulta la sezione Ottenimento delle informazioni sullo stato della replica.

Se non si filtra il processo di replica in batch, Batch Operations tenterà di replicare tutti gli oggetti (indipendentemente dal loro`ObjectReplicationStatus`) nel file manifest che corrispondono alle regole della configurazione di replica, ad eccezione di alcuni oggetti che non vengono replicati per impostazione predefinita. Per ulteriori informazioni, consulta the section called “Che cosa non viene replicato con le configurazioni di replica?”

A seconda dell'obiettivo, è possibile `ObjectReplicationStatuses` impostare uno o più dei seguenti valori:

- Per replicare solo oggetti esistenti che non sono mai stati replicati, includi solo. `"NONE"`
- Per riprovare a replicare solo gli oggetti che in precedenza non erano stati replicati, includi solo. `"FAILED"`
- Per replicare oggetti esistenti e riprovare a replicare oggetti che in precedenza non erano riusciti a replicare, includi entrambi e. `"NONE" "FAILED"`
- Per riempire un bucket di destinazione con oggetti che sono stati replicati in un'altra destinazione, includi. `"COMPLETED"`
- Per replicare oggetti che erano stati replicati in precedenza, includi. `"REPLICA"`

Report di completamento della replica in batch

Quando crei un processo di Batch Replication, puoi richiedere un report di completamento in formato CVS. Questo report mostra gli oggetti, i codici di esito positivo o negativo della replica, gli output e

le descrizioni. Per ulteriori informazioni sul monitoraggio dei lavori e sui report di completamento, vedere. Rapporti di completamento

Per un elenco dei codici e delle descrizioni degli errori di replica, vedereMotivi degli errori di replica Amazon S3.

Per informazioni sulla risoluzione dei problemi relativi alla replica in batch, vedereErrori di replica in batch.

Guida introduttiva alla replica in batch

Per ulteriori informazioni su come utilizzare la replica in batch, consulta il Tutorial: Replicating existing objects in your Amazon S3 buckets with S3 Batch Replication (Replica di oggetti esistenti nei bucket Amazon S3 con S3 Batch Replication).

Configurazione delle policy IAM per Batch Replication

Poiché S3 Batch Replication è un tipo di processo Batch Operations, devi creare un ruolo AWS Identity and Access Management (IAM) per Batch Operations al fine di concedere ad Amazon S3 le autorizzazioni per eseguire operazioni per tuo conto. Inoltre, devi collegare una policy IAM di Batch Replication al ruolo IAM di Batch Operations. Nell'esempio seguente viene creato un ruolo IAM che fornisce a Batch Operations l'autorizzazione per avviare un processo Batch Replication.

Creazione di una policy e un ruolo IAM

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. In Access management (Gestione accessi), scegli Roles (Ruoli).
3. Selezionare Crea ruolo.
4. Scegli Servizio AWS come tipo di entità attendibile, Amazon S3 come servizio e S3 Batch Operations (Operazioni di batch S3) come caso d'uso.
5. Scegli Next: Permissions (Successivo: Autorizzazioni).
6. Selezionare Create Policy (Crea policy).
7. Scegli JSON e inserisci una delle seguenti policy in base al tuo manifesto.

> **Note**
>
> Sono necessarie autorizzazioni diverse in base al tipo di manifesto, che può essere generato o fornito dall'utente. Per ulteriori informazioni, consulta Specifica di un manifesto per un processo Batch Replication.

Politica di utilizzo e archiviazione di un manifesto generato da 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 ****/*"
         ]
      }
   ]
}
```

Policy nel caso in cui impieghi un manifesto fornito dall'utente

```
{
   "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. Scegliere Next: Tags (Successivo: Tag).
9. Scegliere Next:Review (Successivo: Rivedi).
10. Scegli un nome per la policy, quindi scegli Create policy (Crea policy).
11. Collega questa policy al tuo ruolo e scegli Next: Tags (Successivo: tag).
12. Scegli Prossimo: Rivedi.
13. Scegli un nome per il ruolo, quindi scegli Create role (Crea ruolo).

Verifica della policy di attendibilità

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. In Access management (Gestione degli accessi), scegli Roles (Ruoli) e seleziona il ruolo appena creato.
3. Nella scheda Trust relationships (Relazioni di attendibilità), scegli Edit trust relationship (Modifica relazione di attendibilità).
4. Verifica che questo ruolo stia utilizzando la seguente policy di attendibilità:

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

Creazione di un processo Batch Replication per una prima regola di replica o una nuova destinazione

Quando si crea la prima regola in una nuova configurazione di replica o si aggiunge una nuova destinazione a una configurazione esistente tramite AWS Management Console, è possibile facoltativamente creare un processo di replica in batch.

Per utilizzare Batch Replication per una configurazione esistente senza aggiungere una nuova destinazione, consulta Creazione di un processo Batch Replication per le regole di replica esistenti.

Utilizzo della replica in batch per una nuova regola o destinazione di replica tramite AWS Management Console

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket), scegli il nome del bucket che contiene gli oggetti che desideri replicare.
3. Per creare una nuova regola di replica o modificare una regola esistente, scegliere Management (Gestione) e scorrere verso il basso fino a Replication rules (Regole di replica):

   - Per creare una nuova regola di replica, scegliere Create replication rule (Crea regola di replica).

     > **Note**
     >
     > Per esempi su come configurare una regola di replica di base, consulta la sezione Esempi di configurazione della replica in tempo reale.

   - Per modificare una regola di replica esistente, selezionare la regola, quindi scegliere Edit rule (Modifica regola).
4. Creare la nuova regola di replica o modificare la destinazione per la regola di replica esistente e scegliere Save (Salva).

   Dopo aver creato la prima regola in una nuova configurazione di replica o dopo aver modificato una configurazione esistente per aggiungere una nuova destinazione, viene visualizzata una finestra di dialogo Replicate existing objects? (Replicare gli oggetti esistenti?) che offre la possibilità di creare un processo Batch Replication.
5. Se desideri eseguire questo processo ora, scegli Sì, replica gli oggetti esistenti.

Se desideri eseguire questo processo in un secondo momento, scegli No, non replicare gli oggetti esistenti.

6. Creazione del processo S3 Batch Replication. Il processo S3 Batch Replication ha diverse impostazioni:

   Opzione di esecuzione del processo

   Se si desidera che il processo S3 Batch Replication venga eseguito immediatamente, è possibile scegliere Job runs automatically when ready (Il processo viene eseguito automaticamente quando è pronto). Se si desidera eseguire il processo in un secondo momento, selezionare Job waits to be run when ready (Il processo attende di essere eseguito quando è pronto).

   Se scegli Job runs automatically when ready (Quando è pronto, il processo viene eseguito automaticamente), non sarai in grado di creare e salvare un manifesto Batch Operations. Per salvare il manifesto Batch Operations, scegli Job waits to be run when ready (Quando è pronto, il processo attende di essere eseguito).

   Manifesto Batch Operations

   Il manifesto è un elenco di tutti gli oggetti sui quali desideri eseguire l'operazione specificata. Puoi scegliere di salvare il manifesto Batch Operations. Analogamente ai file di inventario S3, il manifesto viene salvato come file CSV e archiviato in un bucket. Per ulteriori informazioni sui manifesti Batch Operations, consulta la sezione Specifica di un manifest.

   Report di completamento

   S3 Batch Operations esegue un'unica attività per ciascun oggetto specificato nel manifesto. I report di completamento offrono un modo semplice per visualizzare i risultati delle attività in un formato consolidato, senza ulteriori operazioni di configurazione. Puoi richiedere un report di completamento per tutte le attività o solo per le attività fallite. Per ulteriori informazioni sui report di completamento, consulta la sezione Rapporti di completamento.

   Autorizzazioni

   Una delle cause più comuni di errori di replica è l'insufficienza delle autorizzazioni nel ruolo fornito AWS Identity and Access Management (IAM). Per ulteriori informazioni sulla creazione di questo ruolo, consulta la sezione Configurazione delle policy IAM per Batch Replication.

7. Scegli Create Batch Operations job (Crea processo Batch Operations).

Creazione di un processo Batch Replication per le regole di replica esistenti

Puoi configurare S3 Batch Replication per una configurazione di replica esistente utilizzando gli AWS SDK, AWS Command Line Interface (AWS CLI) o la console Amazon S3. Per una panoramica di Batch Replication, consulta la sezione Replica di oggetti esistenti con S3 Batch Replication.

Come prerequisito, devi creare un ruolo Batch Operations AWS Identity and Access Management (IAM) per concedere ad Amazon S3 le autorizzazioni per eseguire azioni per tuo conto, vedi. Configurazione delle policy IAM per Batch Replication

Al termine del processo Batch Replication, viene visualizzato un report di completamento. Per ulteriori informazioni su come utilizzare il report per esaminare il processo, consulta la sezione Monitoraggio dei rapporti sullo stato e sul completamento dei processi.

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Scegliere Batch Operations (Operazioni batch) nel riquadro di navigazione della console di Amazon S3.
3. Selezionare Create job (Crea processo).
4. Scegliere la Region (Regione) in cui creare il processo.
5. Seleziona il Manifest format (Formato manifesto). In questo esempio viene illustrato come creare un manifesto basato su una configurazione di replica S3 esistente.

   > **Note**
   >
   > Il manifesto è un elenco di tutti gli oggetti sui quali desideri eseguire l'operazione specificata. Per ulteriori informazioni sui manifesti Batch Operations, consulta la sezione Specifica di un manifest. Se hai preparato un manifesto, scegli S3 inventory report (manifest.json) (Report di inventario S3 (manifest.json)) o CSV. Se gli oggetti nel manifest sono in un bucket con versione, è necessario specificare gli ID versione per gli oggetti. Per ulteriori informazioni sulla creazione di un file manifesto, consulta la sezione Specifica di un manifest.

6. Per creare un manifesto basato sulla configurazione di replica, scegli Create manifest using S3 Replication configuration (Crea manifesto utilizzando la configurazione di replica S3). Quindi, scegli il bucket di origine della configurazione di replica.

7. (Facoltativo) Puoi includere filtri aggiuntivi come la data di creazione dell'oggetto e lo stato della replica. Per esempi su come filtrare in base allo stato della replica, consulta la sezione Specifica di un manifesto per un processo Batch Replication.
8. Per salvare un manifesto, seleziona Save Batch Operations manifest (Salva manifesto Batch Operations).

   a. Se scegli di generare e salvare un manifesto, devi scegliere tra Bucket in this account (Bucket in questo account) oppure Bucket in another Account AWS(Bucket in un altro Account AWS). Specifica il nome del bucket nella casella di testo.

   > **Note**
   >
   > Il manifesto generato deve essere archiviato nello Regione AWS stesso bucket di origine.

   b. Scegli il Tipo di crittografia.
9. (Facoltativo) Fornisci un valore per Description (Descrizione).
10. Modifica il valore Priority (Priorità) del processo, se necessario. Numeri maggiori indicano una priorità superiore. Amazon S3 tenta di eseguire i processi con priorità più elevata prima dei processi con priorità inferiore. Per ulteriori informazioni sulla priorità dei processi, consulta Assegnazione della priorità dei processi.
11. (Facoltativo) Genera un report di completamento. Per generarlo, seleziona Generate completion report (Genera report di completamento).

    Se scegli di generare un report di completamento, devi scegliere se riferire Failed tasks only (Solo attività fallite) o All tasks (Tutte le attività) e fornire un bucket di destinazione per il report.
12. Seleziona un ruolo IAM valido.

> **Note**
>
> Per ulteriori informazioni sulla creazione di un ruolo IAM, consulta la sezione Configurazione delle policy IAM per Batch Replication.

13. (Facoltativo) Aggiungi tag di processo al processo Batch Replication.
14. Seleziona Successivo.
15. Rivedi la configurazione e seleziona Create job (Crea processo).

Utilizzo di AWS CLI con un manifesto S3

L'esempio seguente crea un job S3 Batch Replication utilizzando un manifesto generato da S3 per. Account AWS *111122223333* Questo esempio tenta di replicare oggetti esistenti e oggetti la cui replica in precedenza era fallita. Per informazioni sul filtro in base allo stato della replica, consulta la sezione Specifica di un manifesto per un processo Batch Replication.

```
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**
>
> Il processo deve essere avviato dallo stesso Regione AWS bucket di origine della replica. Il ruolo IAM `role/batch-Replication-IAM-policy` è stato creato in precedenza. Per informazioni, consulta Configurazione delle policy IAM per Batch Replication.

Dopo aver avviato correttamente un processo Batch Replication, viene visualizzato l'ID del processo come risposta. Puoi monitorare il processo utilizzando il seguente comando.

```
aws s3control describe-job --account-id 111122223333 --job-id job-id --region source-
bucket-region
```

Utilizzo di con un manifesto AWS CLI fornito dall'utente

Nell'esempio seguente viene creato un processo S3 Batch Replication tramite un manifesto definito dall'utente per l' Account AWS *111122223333*. Se gli oggetti nel manifesto sono in un bucket con versione, è necessario specificare gli ID versione per gli oggetti. Verrà replicato solo l'oggetto con l'ID versione specificato nel manifesto. Per ulteriori informazioni sulla creazione di un file manifesto, consulta la sezione Specifica di un manifest.

```
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**
>
> Il processo deve essere avviato dallo stesso bucket di origine della Regione AWS replica. Il ruolo IAM `role/batch-Replication-IAM-policy` è stato creato in precedenza. Per informazioni, consulta Configurazione delle policy IAM per Batch Replication.

Dopo aver avviato correttamente un processo Batch Replication, viene visualizzato l'ID del processo come risposta. Puoi monitorare il processo utilizzando il seguente comando.

```
aws s3control describe-job --account-id 111122223333 --job-id job-id --region source-
bucket-region
```

Suddivisione in categorie dello storage utilizzando i tag

Utilizza il tagging degli oggetti per catalogare lo storage. Ogni tag è una coppia chiave-valore.

È possibile aggiungere tag ai nuovi oggetti durante il caricamento oppure è possibile aggiungerli agli oggetti esistenti.

- È possibile associare fino a un massimo di 10 tag a ciascun oggetto. I tag associati a un oggetto devono avere chiavi di tag univoche.
- Una chiave di tag può essere composta da un massimo di 128 caratteri Unicode e i valori di tag possono essere composti da un massimo di 256 caratteri Unicode. I tag di oggetti Amazon S3 sono rappresentati internamente in UTF-16. I caratteri in UTF-16 usano 1 o 2 posizioni.
- La chiave e i valori fanno distinzione tra maiuscole e minuscole.

- Per ulteriori informazioni sulle restrizioni relative ai tag, consulta Restrizioni relative ai tag definite dall'utente nella AWS Billing and Cost Management User Guide. Per le restrizioni di base sui tag, consulta Restrizioni relative ai tag nella Guida per l'utente di Amazon EC2.

Esempi

Considerare i seguenti esempi di tagging:

Example Informazioni PHI

Supponiamo che un oggetto contenga dati sanitari protetti (PHI). È possibile assegnare un tag all'oggetto utilizzando la seguente coppia chiave-valore.

```
PHI=True
```

oppure

```
Classification=PHI
```

Example File di progetto

Supponiamo di archiviare i file di progetto nel bucket S3. È possibile assegnare un tag a questi oggetti mediante una chiave denominata `Project` e un valore, come illustrato di seguito.

```
Project=Blue
```

Example Tag multipli

È possibile aggiungere più tag a un oggetto, come illustrato di seguito.

```
Project=x
Classification=confidential
```

Prefissi e tag dei nomi delle chiavi

I prefissi dei nomi di una chiave dell'oggetto ti permettono anche di categorizzare lo storage. Tuttavia, la categorizzazione basata sui prefissi è monodimensionale. Consideriamo i seguenti nomi delle chiavi degli oggetti:

```
photos/photo1.jpg
project/projectx/document.pdf
project/projecty/document2.pdf
```

Questi nomi di chiavi hanno il prefisso `photos/`, `project/projectx/` e `project/projecty/`. Questi prefissi consentono la categorizzazione monodimensionale, ossia: tutti gli elementi sotto un prefisso costituiscono una categoria. Ad esempio, il prefisso `project/projectx` identifica tutti i documenti relativi al progetto x.

Il tagging rende disponibile un'altra dimensione. Se si desidera che photo1 sia nella categoria project x, è possibile assegnare un tag all'oggetto di conseguenza.

Altri vantaggi

Oltre alla classificazione dei dati, il tagging offre vantaggi quali i seguenti:

- I tag degli oggetti consentono un controllo degli accessi granulare per le autorizzazioni. Ad esempio, è possibile concedere a un utente le autorizzazioni per leggere esclusivamente gli oggetti con tag specifici.
- I tag degli oggetti consentono una gestione granulare del ciclo di vita dell'oggetto, in cui è possibile specificare filtri basati su tag, oltre a prefissi del nome della chiave, in una regola del ciclo di vita.
- L'utilizzo dell'analisi Amazon S3 consente di configurare filtri per raggruppare gli oggetti per l'analisi in base ai tag dell'oggetto, al prefisso del nome della chiave di accesso o in base sia al prefisso che ai tag.
- Puoi anche personalizzare le CloudWatch metriche di Amazon per visualizzare le informazioni tramite filtri di tag specifici. Nelle seguenti sezioni sono fornite maggiori informazioni.

> **Important**
>
> Si possono utilizzare tag per etichettare oggetti contenenti informazioni riservate (ad esempio le informazioni personali (PII) o i dati sanitari protetti (PHI)). Tuttavia, i tag non devono contenere informazioni confidenziali.

Aggiunta di serie di tag oggetto a più oggetti Amazon S3 con una singola richiesta

Per aggiungere set di tag a più di un oggetto Amazon S3 con una sola richiesta, puoi utilizzare le operazioni in batch S3. Fornisci alle operazioni in batch S3 un elenco di oggetti su cui operare. Le

operazioni in batch S3 richiamano la rispettiva API per eseguire l'operazione specificata. Un solo processo di operazioni in batch può eseguire l'operazione specificata su miliardi di oggetti contenenti esabyte di dati.

La funzionalità S3 Batch Operations tiene traccia dell'avanzamento, invia notifiche e archivia un report di completamento dettagliato di tutte le operazioni, offrendo un'esperienza serverless revisionabile completamente gestita. Puoi utilizzare S3 Batch Operations tramite la console Amazon S3 AWS CLI AWS , gli SDK o l'API REST. Per ulteriori informazioni, consulta the section called “Nozioni di base sulle operazioni in batch”.

Per ulteriori informazioni sui tag degli oggetti, consulta Gestione di tag degli oggetti.

Operazioni API correlate al tagging oggetti

Amazon S3 supporta le seguenti operazioni API, specifiche per il tagging oggetti:

Operazioni delle API sugli oggetti

- PUT Object tagging – Sostituisce i tag su un oggetto. È possibile specificare i tag nel corpo della richiesta. La gestione di tag degli oggetti mediante queste API prevede due scenari distinti.
  - L'oggetto non ha tag – Mediante questa API, è possibile aggiungere un set di tag a un oggetto (l'oggetto non ha tag precedenti).
  - L'oggetto ha un set tag esistenti – Per modificare il set di tag esistenti, è necessario prima recuperarlo, modificarlo sul lato client, quindi utilizzare questa API per sostituire il set di tag.

    > **Note**
    >
    > Se si invia questa richiesta con un set di tag vuoto, Amazon S3 elimina il set di tag esistenti sull'oggetto. Se si usa questo metodo, verrà addebitata una richiesta Tier 1 (PUT). Per ulteriori informazioni, consulta Prezzi di Amazon S3.
    > La richiesta DELETE Object tagging è preferibile perché fornisce lo stesso risultato senza nessun addebito.

- GET Object tagging – Restituisce il set di tag associato a un oggetto. Amazon S3 restituisce i tag degli oggetti nel corpo della risposta.
- DELETE Object tagging – Elimina il set di tag associato a un oggetto.

Altre operazioni API che supportano il tagging

- PUT Object e Initiate Multipart Upload – È possibile specificare i tag quando si creano oggetti. I tag possono essere specificati utilizzando l'intestazione di richiesta `x-amz-tagging`.
- GET Object – Anziché restituire il set di tag, Amazon S3 restituisce il conteggio dei tag degli oggetti nell'intestazione di `x-amz-tag-count` (solo se il richiedente dispone delle autorizzazioni per leggere i tag) poiché le dimensioni della risposta nell'intestazione sono limitate a 8 K di byte. Se si desidera visualizzare i tag, fare un'altra richiesta di operazione API GET Object tagging.
- POST Object – È possibile specificare tag nella richiesta POST.

  È possibile utilizzare l'API `PUT Object` per creare oggetti con tag, purché i tag della richiesta non superino le dimensioni massime dell'intestazione della richiesta HTTP di 8 Kbyte, Se i tag specificati superano le dimensioni massime dell'intestazione, è possibile utilizzare questo metodo POST che consiste nell'includere i tag nel corpo.

  PUT Object - Copy – È possibile specificare `x-amz-tagging-directive` nella richiesta per indicare ad Amazon S3 di copiare (comportamento di default) i tag o sostituirli mediante un nuovo set di tag fornito nella richiesta.

Tieni presente quanto segue:

- Il tagging degli oggetti S3 è molto coerente. Per ulteriori informazioni, consulta Modello di consistenza dati Amazon S3.

Configurazioni aggiuntive

In questa sezione viene descritto in che modo il tagging oggetti è correlato alle altre configurazioni.

Tagging oggetti e gestione del ciclo di vita

In una configurazione del ciclo di vita del bucket, è possibile specificare un filtro per selezionare un sottoinsieme di oggetti a cui si applica la regola. È possibile specificare un filtro in base ai prefissi dei nomi delle chiavi, ai tag degli oggetti o entrambi.

Supponiamo di archiviare foto (in formato raw e in formato finito) nel bucket Amazon S3. A questi oggetti possono essere assegnati tag nel modo seguente.

```
phototype=raw
or
phototype=finished
```

È possibile archiviare le foto in formato raw in S3 Glacier poco dopo la creazione. È possibile configurare una regola del ciclo di vita con un filtro che identifica il sottoinsieme di oggetti con prefisso del nome della chiave (`photos/`) aventi un tag specifico (`phototype=raw`).

Per ulteriori informazioni, consulta Gestione del ciclo di vita dello storage.

Tagging degli oggetti e replica

Se è stata configurata la replica nel bucket, Amazon S3 replica i tag, purché ad Amazon S3 siano assegnate le autorizzazioni per leggerli. Per ulteriori informazioni, consulta Configurazione della replica in tempo reale.

Notifiche eventi di assegnazione tag su oggetti

Puoi configurare una notifica eventi Amazon S3 per ricevere una notifica quando viene aggiunto o eliminato un tag oggetto da un oggetto. Il tipo di evento `s3:ObjectTagging:Put` ti avvisa quando un tag viene INSERITO su un oggetto o quando viene aggiornato un tag esistente. Il tipo di evento `s3:ObjectTagging:Delete` ti avvisa quando un tag viene rimosso da un oggetto. Per ulteriori informazioni, consulta Abilitazione notifiche eventi.

Per ulteriori informazioni sul tagging degli oggetti, consulta i seguenti argomenti:

Argomenti

- Tagging e policy di controllo degli accessi
- Gestione di tag degli oggetti

Tagging e policy di controllo degli accessi

Le policy di autorizzazione (policy bucket e policy utente) possono essere utilizzate per gestire le autorizzazioni relative al tagging oggetti. Per le operazioni delle policy, consulta i seguenti argomenti:

- Operazioni sugli oggetti
- Operazioni relative ai bucket

I tag degli oggetti consentono un controllo degli accessi granulare per la gestione delle autorizzazioni. È possibile concedere autorizzazioni condizionali in base ai tag degli oggetti. Amazon S3 supporta le seguenti chiavi di condizione che è possibile utilizzare per concedere autorizzazioni condizionali basate sui tag degli oggetti.

- `s3:ExistingObjectTag/<tag-key>` – Utilizzare questa chiave di condizione per verificare che un tag degli oggetti esistente abbia una chiave e un valore di tag specifici.

  > **Note**
  >
  > Quando si concedono autorizzazioni per le operazioni `PUT Object` e `DELETE Object`, questa chiave di condizione non è supportata. Ciò significa che non è possibile creare una policy per concedere o rifiutare le autorizzazioni utente che consentono di eliminare o sovrascrivere un oggetto in base ai relativi tag esistenti.

- `s3:RequestObjectTagKeys` – Utilizzare questa chiave di condizione per limitare le chiavi di tag che si desidera consentire sugli oggetti. Ciò è utile quando si aggiungono tag agli oggetti utilizzando le richieste di oggetti PutObjectTagging and PutObject e POST.
- `s3:RequestObjectTag/<tag-key>` – Utilizzare questa chiave di condizione per limitare i valori e le chiavi di tag che si desidera consentire sugli oggetti. Ciò è utile quando si aggiungono tag agli oggetti utilizzando le richieste PutObjectTagging and PutObject e POST Bucket.

Per un elenco completo delle chiavi di condizione specifiche per il servizio Amazon S3, consulta Esempi di policy Bucket che utilizzano chiavi condizionali. Le seguenti policy di autorizzazione illustrano il modo in cui il tagging oggetti consente una gestione granulare delle autorizzazioni di accesso.

Example 1: concedere a un utente autorizzazioni di sola lettura per gli oggetti con un valore di tag o chiave specifico

La seguente policy di autorizzazione limita un utente a leggere solo gli oggetti con chiave e valore di tag `environment: production`. La policy utilizza la chiave di condizione `s3:ExistingObjectTag` per specificare la chiave e il valore di tag.

```
{
  "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: limitare le chiavi di tag dell'oggetto che gli utenti possono aggiungere

La seguente policy di autorizzazione concede a un utente le autorizzazioni per eseguire l'operazione `s3:PutObjectTagging`, che permette di aggiungere tag a un oggetto esistente. La condizione utilizza la chiave di condizione `s3:RequestObjectTagKeys` per specificare le chiavi di tag consentite, ad esempio `Owner` o `CreationDate`. Per ulteriori informazioni, consulta la sezione Creazione di una condizione con più chiavi o valori nella Guida per l'utente IAM.

La policy garantisce che ogni chiave di tag specificata nella richiesta sia una chiave di tag autorizzata. Il qualificatore `ForAnyValue` nella condizione garantisce che almeno una delle chiavi specificate sia presente nella richiesta.

```
{
   "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: richiedere una chiave e un valore di tag specifici per consentire agli utenti di aggiungere tag di oggetti

La seguente policy di esempio concede a un utente le autorizzazioni per eseguire l'operazione `s3:PutObjectTagging`, che permette di aggiungere tag a un oggetto esistente. La condizione prevede che l'utente includa una chiave di tag specifica (ad esempio, *`Project`*) con valore impostato su *`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"
        }
      }
    }
  ]
}
```

Gestione di tag degli oggetti

Questa sezione spiega come gestire i tag degli oggetti utilizzando gli AWS SDK per Java e.NET o la console Amazon S3.

Il tagging ti consente di catalogare lo storage. Ciascun tag è una coppia chiave-valore che aderisce alle seguenti regole:

- È possibile associare fino a un massimo di 10 tag a ciascun oggetto. I tag associati a un oggetto devono avere chiavi di tag univoche.
- Una chiave di tag può essere composta da un massimo di 128 caratteri Unicode e i valori di tag possono essere composti da un massimo di 256 caratteri Unicode. I tag di oggetti Amazon S3 sono rappresentati internamente in UTF-16. I caratteri in UTF-16 usano 1 o 2 posizioni.
- La chiave e i valori fanno distinzione tra maiuscole e minuscole.

Per ulteriori informazioni sui tag degli oggetti, consulta Suddivisione in categorie dello storage utilizzando i tag. Per ulteriori informazioni sui limiti dei tag, consulta Restrizioni sui tag definiti dall'utente nella Guida per l'utente AWS Billing and Cost Management .

Utilizzo della console S3

Per aggiungere tag a un oggetto

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket, scegli il nome del bucket che contiene gli oggetti a cui desideri aggiungere i tag.

   Se necessario, puoi passare a una cartella.
3. Nell'elenco Oggetti, seleziona la casella di controllo accanto ai nomi degli oggetti a cui desideri aggiungere i tag.
4. Dal menu Operazioni, seleziona Modifica tag.
5. Esamina gli oggetti elencati e seleziona Aggiungi tag.
6. Ogni tag oggetto è una coppia chiave-valore. Immettere una chiave e un valore. Per aggiungere un altro tag, scegliere Add Tag (Aggiungi tag).

   È possibile immettere fino a un massimo di 10 tag per ciascun oggetto.
7. Seleziona Salva modifiche.

   Amazon S3 aggiungerà i tag agli oggetti specificati.

Per ulteriori informazioni, vedi anche Visualizzazione delle proprietà di un oggetto nella console di Amazon S3 e Caricamento degli oggetti in questa guida.

Utilizzo degli SDK AWS

Java

L'esempio seguente mostra come utilizzare per impostare i AWS SDK for Java tag per un nuovo oggetto e recuperare o sostituire i tag per un oggetto esistente. Per ulteriori informazioni sul tagging dell'oggetto, consulta Suddivisione in categorie dello storage utilizzando i tag. Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

public class ManagingObjectTags {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Object key ***";
        String filePath = "*** File path ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Create an object, add two new tags, and upload the object to Amazon
 S3.
            PutObjectRequest putRequest = new PutObjectRequest(bucketName, keyName,
 new File(filePath));
            List<Tag> tags = new ArrayList<Tag>();
            tags.add(new Tag("Tag 1", "This is tag 1"));
```

```
            tags.add(new Tag("Tag 2", "This is tag 2"));
            putRequest.setTagging(new ObjectTagging(tags));
            PutObjectResult putResult = s3Client.putObject(putRequest);

            // Retrieve the object's tags.
            GetObjectTaggingRequest getTaggingRequest = new
 GetObjectTaggingRequest(bucketName, keyName);
            GetObjectTaggingResult getTagsResult =
 s3Client.getObjectTagging(getTaggingRequest);

            // Replace the object's tags with two new tags.
            List<Tag> newTags = new ArrayList<Tag>();
            newTags.add(new Tag("Tag 3", "This is tag 3"));
            newTags.add(new Tag("Tag 4", "This is tag 4"));
            s3Client.setObjectTagging(new SetObjectTaggingRequest(bucketName,
 keyName, new ObjectTagging(newTags)));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

L'esempio seguente mostra come utilizzare AWS SDK for .NET per impostare i tag per un nuovo oggetto e recuperare o sostituire i tag per un oggetto esistente. Per ulteriori informazioni sul tagging dell'oggetto, consulta Suddivisione in categorie dello storage utilizzando i tag.

Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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);
            }
        }
    }
}
```

Utilizzo dei tag per l'allocazione dei costi per i bucket S3

Per monitorare i costi di storage o altri criteri per singoli progetti o gruppi di progetti, etichettare i bucket Amazon S3 mediante i tag di allocazione dei costi. Un tag di allocazione dei costi è una coppia chiave/valore associata a un bucket S3. Una volta attivati, i tag per l'allocazione dei costi vengono utilizzati da AWS per organizzare i costi delle risorse nel report di allocazione dei costi. I tag per l'allocazione dei costi possono essere utilizzati solo per etichettare i bucket. Per informazioni sui tag utilizzati per etichettare gli oggetti, consulta Suddivisione in categorie dello storage utilizzando i tag.

Il report mensile di allocazione dei costi elenca l'utilizzo di AWS per il tuo account per categoria di prodotto e utente dell'account collegato. Il report contiene le stesse voci di utilizzo indicate nel report di fatturazione dettagliato (vedere Comprendere i report AWS di fatturazione e utilizzo per Amazon S3) e altre colonne relative alle chiavi dei tag.

AWS fornisce due tipi di tag di allocazione dei costi, un tag generato da AWS e dei tag definiti dall'utente. AWS definisce, crea e applica il tag `createdBy` generato da AWS dopo un evento CreateBucket di Amazon S3. Tu puoi definire, creare e applicare tag definiti dall'utente al bucket S3.

È necessario attivare entrambi i tipi di tag separatamente nella console Gestione di costi e fatturazione prima di poterli visualizzare nei report di fatturazione. Per ulteriori informazioni sui tag generati da AWS, consulta la pagina relativa ai tag di allocazione dei costi generati da AWS.

- Per creare tag nella console, consulta Visualizzazione delle proprietà di un bucket S3.
- Per creare tag utilizzando l'API Amazon S3, consulta l'argomento relativo ai tagging PUT Bucket nella Documentazione di riferimento delle API di Amazon Simple Storage Service.
- Per creare tag utilizzando AWS CLI, consulta put-bucket-tagging in Riferimento ai comandi della CLI AWS CLI.

- Per ulteriori informazioni sull'attivazione dei tag, consulta Utilizzo dei tag per l'allocazione dei costi nella Guida per l'utente di AWS Billing.

Tag per l'allocazione dei costi definiti dall'utente

Un tag per l'allocazione dei costi definito dall'utente presenta i seguenti componenti:

- La chiave di tag: La chiave di tag corrisponde al nome del tag. Ad esempio, nel tag project/Trinity, project è la chiave. La chiave di tag è una stringa che fa distinzione tra maiuscole e minuscole, contenente da 1 a 128 caratteri Unicode.

- Il valore del tag. Il valore del tag è una stringa obbligatoria. Ad esempio, nel tag project/Trinity, Trinity è il valore. Il valore del tag è una stringa che fa distinzione tra maiuscole e minuscole, contenente da 0 a 256 caratteri Unicode.

Per informazioni sui caratteri consentiti per i tag definiti dall'utente e altre limitazioni, consulta Limitazioni per i tag definiti dall'utente nella Guida per l'utente di AWS Billing. Per ulteriori informazioni sui tag definiti dall'utente, consulta Tag per l'allocazione dei costi definiti dall'utente nella Guida per l'utente di AWS Billing.

Tag bucket S3

Ogni bucket S3 dispone di un set di tag. Un set di tag contiene tutti i tag assegnati al bucket. Un set di tag può contenere fino a 50 tag o può essere vuoto. Nell'ambito di un set di tag, le chiavi devono essere univoche ma non occorre che i valori di un set di tag siano univoci. Ad esempio, si può avere lo stesso valore nei set di tag denominati project/Trinity e cost-center/Trinity.

Nell'ambito di un bucket, se si aggiunge un tag la cui chiave è la stessa di un tag esistente, il nuovo valore sovrascrive il vecchio valore.

AWS non applica significati semantici ai tag. I tag sono interpretati prettamente come stringhe di caratteri.

Per aggiungere, elencare, modificare o eliminare tag, è possibile utilizzare la console di Amazon S3, AWS Command Line Interface (AWS CLI) o l'API Amazon S3.

Ulteriori informazioni

- Uso dei tag per l'allocazione dei costi nella Guida per l'utente di AWS Billing.

- Comprendere i report AWS di fatturazione e utilizzo per Amazon S3
- AWS Billing report per Amazon S3

Report di fatturazione e utilizzo per Amazon S3

> **Important**
>
> Il 13 maggio 2024, abbiamo iniziato a implementare una modifica per eliminare gli addebiti per le richieste non autorizzate non avviate dal proprietario del bucket. Una volta completata l'implementazione di questa modifica, i proprietari dei bucket non dovranno mai sostenere costi di richiesta o larghezza di banda per le richieste che restituiscono errori `AccessDenied` (HTTP`403 Forbidden`) quando tali richieste vengono avviate dall'esterno del loro account o organizzazione individuale. AWS AWS Per ulteriori informazioni sull'elenco completo dei codici HTTP `3XX` e di `4XX` stato che non verranno fatturati, consulta. Risposte agli errori di fatturazione per Amazon S3 Questa modifica alla fatturazione non richiede aggiornamenti alle applicazioni e si applica a tutti i bucket S3. Una volta completata l'implementazione di questa modifica Regioni AWS, aggiorneremo la nostra documentazione.

Quando usi Amazon S3, non devi pagare alcuna commissione anticipata o impegnarti a gestire la quantità di contenuti da archiviare. Come tutti gli altri Servizi AWS, paghi solo in base all'uso effettivo.

AWS fornisce i seguenti report per Amazon S3:

- Report di fatturazione: report multipli che forniscono viste di alto livello di tutte le attività del sistema Servizi AWS che stai utilizzando, incluso Amazon S3. AWS fattura sempre al proprietario del bucket S3 le tariffe di Amazon S3, a meno che il bucket non sia stato creato come bucket Requester Pays. Per ulteriori informazioni sui tipi di Pagamento a carico del richiedente, consulta Utilizzo dei bucket con pagamento a carico del richiedente per utilizzo e trasferimenti di storage. Per ulteriori informazioni sui report di fatturazione, consulta AWS Billing report per Amazon S3.
- Report di utilizzo: un riepilogo delle attività relative a uno specifico servizio, raggruppate per ora, giorno o mese. È possibile scegliere quale tipo di utilizzo e operazione includere. È inoltre possibile scegliere la modalità di aggregazione dei dati. Per ulteriori informazioni, consulta AWS rapporto di utilizzo per Amazon S3.

I seguenti argomenti forniscono informazioni sulla creazione di report di utilizzo e fatturazione per Amazon S3.

Argomenti

AWS Billing report per Amazon S3

La fattura mensile AWS separa le informazioni sull'utilizzo e i costi per funzione Servizio AWS . Sono disponibili diversi AWS Billing report: il rapporto mensile, il rapporto sull'allocazione dei costi e i report di fatturazione dettagliati. Per informazioni su come visualizzare i report di fatturazione, consulta Visualizzazione di una fattura nella Guida per l'utente di AWS Billing .

Per monitorare AWS l'utilizzo e fornire una stima degli addebiti associati al tuo account, puoi configurare. AWS Cost and Usage Reports Per ulteriori informazioni, consulta Cosa sono AWSCost and Usage Reports? nella Guida all'esportazione AWS dei dati.

È inoltre possibile scaricare un report di utilizzo che fornisce informazioni più dettagliate sull'utilizzo dello storage Amazon S3 rispetto ai report di fatturazione. Per ulteriori informazioni, consulta AWS rapporto di utilizzo per Amazon S3.

Nella tabella seguente sono riportati i costi associati all'utilizzo di Amazon S3.

Costi di utilizzo di Amazon S3

Costo	Commenti
Storage	L'archiviazione degli oggetti nei bucket S3 è a pagamento. La tariffa che ti viene addebitat a dipende dalla dimensione degli oggetti, dal tempo di archiviazione degli oggetti durante il mese e dalla classe di archiviazione. Amazon S3 offre le seguenti classi di storage: S3 Standard, S3 Express One Zone, S3 Intellige nt-Tiering, S3 Standard-IA (IA per accesso non

Costo	Commenti
	frequente), S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive o Reduced Redundancy Storage (RRS). Per ulteriori informazioni sulle classi di storage, consulta Utilizzo delle classi di storage di Amazon S3. Tieni presente che se hai abilitato S3 Versionin g, ti verrà addebitato un costo per ogni versione di un oggetto che viene conservata. Per ulteriori informazioni sulla funzione Controllo delle versioni, consulta Come funzionano il Controllo delle versioni S3.
Monitoraggio e automazione	Sarà applicata una commissione mensile per il monitoraggio e l'automazione per ciascun oggetto archiviato nella classe di storage S3 Intelligent-Tiering per monitorare i pattern di accesso e per spostare gli oggetti tra livelli di accesso nell'S3 Intelligent-Tiering.
Richieste	Paghi per le richieste, ad esempio le `GET` richieste, effettuate sui tuoi bucket e oggetti S3. Sono incluse le richieste del ciclo di vita. Le tariffe per le richieste dipendono dal tipo di richiesta che stai effettuando. Per informazi oni sui prezzi delle richieste, consulta Prezzi di Amazon S3.
Recuperi	Il recupero degli oggetti archiviati in S3 Standard-IA, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive è a pagamento.

Costo	Commenti
Eliminazioni anticipate	Se elimini un oggetto archiviato nelle classi di archiviazione S3 Standard-IA, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive prima del termine del periodo minimo di archiviazione, ti verrà addebitato il costo per l'eliminazione prematura dell'oggetto.
Gestione dello storage	Paghi per le funzionalità di gestione dello storage (Amazon S3 Inventory, analisi e etichettatura degli oggetti) abilitate nei bucket del tuo account.
Larghezza di banda	Ti sarà addebitato l'intero costo della larghezza di banda in entrata e in uscita da Amazon S3, eccetto nei seguenti casi: • Dati trasferiti in entrata da Internet • Dati trasferiti su un'istanza Amazon Elastic Compute Cloud (Amazon EC2), quando l'istanza si trova nello stesso bucket S3 Regione AWS • Dati trasferiti su Amazon CloudFront (CloudFront) Inoltre, paghi una tariffa per tutti i dati trasferiti utilizzando Amazon S3 Transfer Acceleration.

Per informazioni dettagliate sui costi di utilizzo di Amazon S3 per lo storage, il trasferimento di dati e i servizi, consulta i prezzi di Amazon S3 e le domande frequenti su Amazon S3.

Per informazioni sulla comprensione dei codici e delle abbreviazioni utilizzati nei report di fatturazione e utilizzo per Amazon S3, consulta. Comprendere i report AWS di fatturazione e utilizzo per Amazon S3

Ulteriori informazioni

- AWS rapporto di utilizzo per Amazon S3
- Utilizzo dei tag per l'allocazione dei costi per i bucket S3
- AWS Billing e gestione dei costi
- Prezzi di Amazon S3

AWS rapporto di utilizzo per Amazon S3

Quando si scarica un report di utilizzo, è possibile scegliere di aggregare i dati sull'utilizzo in base a ora, giorno o mese. Il report sull'utilizzo di Amazon S3 elenca le operazioni per tipo di utilizzo e. Regione AWS Per informazioni più dettagliate sull'utilizzo dell'archiviazione Amazon S3, scarica i report di utilizzo AWS generati in modo dinamico. È possibile scegliere quale tipo di utilizzo, operazione e periodo di tempo includere. È inoltre possibile scegliere la modalità di aggregazione dei dati. Per ulteriori informazioni sui report di utilizzo, consulta il Report di AWS utilizzo nella AWS Data Exports User Guide.

Il report di utilizzo di Amazon S3 include le seguenti informazioni:

- Servizio - Amazon S3
- Operation (Operazione) – Operazione eseguita sul bucket o sull'oggetto. Per una spiegazione dettagliata delle operazioni di Amazon S3, consulta Operazioni di monitoraggio nei report di utilizzo.
- UsageType— Uno dei seguenti valori:
  - Un codice che identifica il tipo di storage
  - Un codice che identifica il tipo di richiesta
  - Un codice che identifica il tipo di recupero
  - Un codice che identifica il tipo di trasferimento dei dati
  - Un codice che identifica l'eliminazione anticipata dall'archivio 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` – Numero di oggetti archiviati in un determinato bucket

  Per una spiegazione dettagliata dei tipi di utilizzo di Amazon S3, consulta Comprendere i report AWS di fatturazione e utilizzo per Amazon S3.

- Resource (Risorsa) – Nome del bucket associato all'utilizzo indicato.
- StartTime— Ora di inizio del giorno a cui si applica l'utilizzo, in UTC (Coordinated Universal Time).
- EndTime— Ora di fine del giorno a cui si riferisce l'utilizzo, in UTC (Coordinated Universal Time).
- UsageValue— Uno dei seguenti valori di volume. L'unità di misura tipica per i dati è gigabyte (GB). Tuttavia, a seconda del servizio e del report, potrebbero invece essere visualizzati terabyte (TB).
  - Il numero di richieste durante il periodo di tempo specificato
  - La quantità di dati trasferiti
  - La quantità di dati memorizzati in una data ora
  - La quantità di dati associata alle operazioni di ripristino dall'archivio S3 Standard-IA, S3 One Zone-IA, S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive

> **Tip**
>
> Per informazioni dettagliate su ogni richiesta ricevuta da Amazon S3 per gli oggetti, attivare la registrazione degli accessi al server per i bucket. Per ulteriori informazioni, consulta Registrazione delle richieste con registrazione dell'accesso al server.

È possibile scaricare un report di utilizzo in formato XML o CSV (Comma-Separated Value). Di seguito è riportato un esempio di report di utilizzo in formato CSV aperto in un foglio elettronico.

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

Per ulteriori informazioni, consulta Comprendere i report AWS di fatturazione e utilizzo per Amazon S3.

Scaricamento del rapporto sull'utilizzo AWS

Puoi scaricare un rapporto sull'utilizzo come file XML o CSV.

Per scaricare il report di utilizzo

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nella barra del titolo, scegli il tuo nome utente o ID account, quindi scegli Billing and Cost Management.
3. Nel riquadro di navigazione, scegli Rapporti su costi e utilizzo.
4. In Rapporto AWS sull'utilizzo, scegli Crea un rapporto sull'utilizzo.
5. Nella pagina Scarica il rapporto sull'utilizzo, scegli le seguenti impostazioni:
   - Servizi: scegli Amazon Simple Storage Service.
   - Usage Types (Tipi di utilizzo) – Per una spiegazione dettagliata dei tipi di utilizzo di Amazon S3, consulta Comprendere i report AWS di fatturazione e utilizzo per Amazon S3.
   - Operation (Operazione) – Per una spiegazione dettagliata delle operazioni di Amazon S3, consulta Operazioni di monitoraggio nei report di utilizzo.
   - Time Period (Periodo di tempo) – Periodo di tempo per cui il report deve fornire informazioni.
   - Report Granularity (Granularità del report) – Consente di specificare se si desidera che il report includa subtotali in base all'ora, al giorno o al mese.
6. Scegli Download, scegli il formato di download (report XML o report CSV), quindi segui le istruzioni per aprire o salvare il report.

Ulteriori informazioni

- Comprendere i report AWS di fatturazione e utilizzo per Amazon S3
- AWS Billing report per Amazon S3

Comprendere i report AWS di fatturazione e utilizzo per Amazon S3

> **Important**
>
> Il 13 maggio 2024, abbiamo iniziato a implementare una modifica per eliminare gli addebiti per le richieste non autorizzate non avviate dal proprietario del bucket. Una volta completata l'implementazione di questa modifica, i proprietari dei bucket non dovranno mai sostenere costi di richiesta o larghezza di banda per le richieste che restituiscono errori `AccessDenied` (HTTP`403 Forbidden`) quando tali richieste vengono avviate dall'esterno del loro account o organizzazione individuale. AWS AWS Per ulteriori informazioni sull'elenco completo dei codici HTTP `3XX` e di `4XX` stato che non verranno fatturati, consulta. Risposte agli errori di fatturazione per Amazon S3 Questa modifica alla fatturazione non richiede aggiornamenti alle applicazioni e si applica a tutti i bucket S3. Una volta completata l'implementazione di questa modifica Regioni AWS, aggiorneremo la nostra documentazione.

I report di utilizzo e fatturazione di Amazon S3 usano codici e abbreviazioni. Per i tipi di utilizzo riportati nella tabella seguente *`regionregion1`*, sostituisci e *`region2`* con le abbreviazioni di questo elenco:

- APE1: Asia Pacifico (Hong Kong)
- APN1: Asia Pacifico (Tokyo)
- APN2: Asia Pacifico (Seoul)
- APN3: Asia Pacifico (Osaka)
- APS1: Asia Pacifico (Singapore)
- APS2: Asia Pacifico (Sydney)
- APS3: Asia Pacifico (Mumbai)
- APS4: Asia Pacifico (Giacarta)
- APS5: Asia Pacifico (Hyderabad)
- APS6: Asia Pacifico (Melbourne)
- CAN1: Canada (Centrale)
- CAN2: Canada occidentale (Calgary)
- CNN1: Pechino (Pechino)
- CNW1: Cina (Ningxia)

- AFS1: Africa (Città del Capo)
- EUC2: Europa (Zurigo)
- EUN1: Europa (Stoccolma)
- EUS2: Europa (Spagna)
- EUC1: Europa (Francoforte)
- EU: Europa (Irlanda)
- EUS1: Europa (Milano)
- EUW2: Europa (Londra)
- EUW3: Europa (Parigi)
- ILC1: Israele (Tel Aviv)
- MEC1: Medio Oriente (Emirati Arabi Uniti)
- MES1: Medio Oriente (Bahrein)
- SAE1: Sud America (San Paolo)
- UGW1: (Stati Uniti occidentali) AWS GovCloud
- UGE 1: (Stati Uniti orientali) AWS GovCloud
- USE1 (o nessun prefisso): Stati Uniti orientali (Virginia settentrionale)
- USE2: Stati Uniti orientali (Ohio)
- USW1: Stati Uniti occidentali (California settentrionale)
- USW2: Stati Uniti occidentali (Oregon)

Per i tipi di utilizzo dei punti di accesso multiregionali S3 riportati nella tabella seguente, sostituisci *regiongroup1* e *regiongroup2* con le abbreviazioni presenti in questo elenco:

- AP: Asia Pacifico
- AU: Australia
- EU: Europa
- IN: India
- NA: Nord America
- SA: Sud America

I gruppi di regioni sono raggruppamenti geografici di più regioni. Regioni AWS Per ulteriori informazioni, consulta Regioni e zone di disponibilità. Per informazioni sui prezzi per Regione AWS, consulta Prezzi di Amazon S3.

La prima colonna della tabella che segue elenca i tipi di utilizzo presenti nei report di utilizzo e fatturazione. L'unità di misura tipica per i dati è gigabyte (GB). Tuttavia, a seconda del servizio e del report, potrebbero invece essere visualizzati terabyte (TB).

Tipi di utilizzo

Tipo di utilizzo	unità	Granularità	Descrizione
region1-*region2*-AWS-In-A Bytes	GB	Orario	La quantità di dati accelerati trasferiti da *region1* *region2*
region1-*region2*-AWS-In-A Bytes-T1	GB	Orario	La quantità di dati accelerati T1 trasferiti *region1* da*region2*, dove T1 si riferisce CloudFront alle richieste ai punti di presenza (POPs) negli Stati Uniti, in Europa e in Giappone
region1-*region2*-AWS-In-A Bytes-T2	GB	Orario	La quantità di dati accelerati T2 trasferiti *region1* da*region2*, dove T2 si riferisce alle CloudFront richieste in tutte le altre location periferiche POPs AWS
region1-*region2*-AWS-In-Bytes	GB	Orario	La quantità di dati trasferit i da *region1* *region2*
region1-*region2*-AWS-Out-ABytes	GB	Orario	La quantità di dati accelerati trasferiti da a *region1* *region2*

Tipo di utilizzo	unità	Granularità	Descrizione
region1-*region2*-AWS-Out-ABytes-T1	GB	Orario	La quantità di dati accelerati T1 trasferiti da *region1* a*region2*, dove T1 si riferisce CloudFront alle richieste ai POP negli Stati Uniti, in Europa e in Giappone
region1-*region2*-AWS-Out-ABytes-T2	GB	Orario	La quantità di dati accelerati T2 trasferiti da *region1* a*region2*, dove T2 si riferisce alle CloudFront richieste ai POP in tutte le altre sedi periferiche AWS
region1-*region2*-AWS-Out-Bytes	GB	Orario	La quantità di dati trasferiti da a *region1* *region2*
region-BatchOperations-Jobs	Conteggio	Orario	Il numero di processi di S3 Batch Operations eseguiti
region-BatchOperations-Objects	Conteggio	Orario	Il numero di operazioni sugli oggetti eseguite da S3 Batch Operations
region-Bulk-Retrieval-Bytes	GB	Orario	La quantità di dati recuperati con richieste S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive di tipo Bulk

Tipo di utilizzo	unità	Granularità	Descrizione
region-BytesDeleted-GDA	GB	Mensile	La quantità di dati eliminati da un'DeleteObject operazione dallo storage S3 Glacier Deep Archive
region-BytesDeleted-GIR	GB	Mensile	La quantità di dati eliminati da un'DeleteObject operazione dallo storage S3 Glacier Instant Retrieval.
region-BytesDeleted-GLACIER	GB	Mensile	La quantità di dati eliminati da un'DeleteObject operazione dallo storage S3 Glacier Flexible Retrieval
region-BytesDeleted-INT	GB	Mensile	La quantità di dati eliminati da un'operazione dallo storage S3 DeleteObject Intelligent-Tiering
region-BytesDeleted-RRS	GB	Mensile	La quantità di dati eliminati da un'DeleteObject operazione dallo storage RRRS (Reduced Redundancy Storage)

Tipo di utilizzo	unità	Granularità	Descrizione
region-BytesDeleted-SIA	GB	Mensile	La quantità di dati eliminati da un'DeleteObject operazione dallo storage S3 Standard-IA
region-BytesDeleted-STANDARD	GB	Mensile	La quantità di dati eliminati da un'DeleteObject operazione dallo storage S3 Standard
region-BytesDeleted-ZIA	GB	Mensile	La quantità di dati eliminati da un'DeleteObject operazione dallo storage S3 One Zone-IA
region-C3DataTransfer-In-Bytes	GB	Orario	La quantità di dati trasferiti in Amazon S3 da Amazon EC2 all'interno dello stesso Regione AWS
region-C3DataTransfer-Out-Bytes	GB	Orario	La quantità di dati trasferiti da Amazon S3 ad Amazon EC2 nell'ambito della stessa Regione AWS
region-CloudFront-In-Bytes	GB	Orario	La quantità di dati trasferiti Regione AWS da e verso una distribuzione CloudFront

Tipo di utilizzo	unità	Granularità	Descrizione
region-CloudFront-Out-Bytes	GB	Orario	La quantità di dati trasferiti da una Regione AWS a una CloudFront distribuzione
region-DataTransfer-In-Bytes	GB	Orario	Quantità di dati trasferiti in Amazon S3 da Internet
region-DataTransfer-Out-Bytes	GB	Orario	Quantità di dati trasferiti da Amazon S3 a Internet[1]
region-DataTransfer-Regional-Bytes	GB	Orario	La quantità di dati trasferiti da Amazon S3 alle AWS risorse all'interno dello stesso Regione AWS
region-EarlyDelete-ByteHrs	GB/ora	Orario	Utilizzo dell'archiviazione ripartita proporzionalmente per gli oggetti eliminati dall'archivio S3 Glacier Flexible Retrieval prima del termine minimo di 90 giorni[2]
region-EarlyDelete-GDA	GB/ora	Orario	Utilizzo dello storage ripartito proporzionalmente per gli oggetti eliminati dallo storage S3 Glacier Deep Archive prima del termine minimo di 180 giorni [2]

Tipo di utilizzo	unità	Granularità	Descrizione
region-EarlyDelete-GIR	GB/ora	Orario	Utilizzo dell'archiviazione ripartita proporzionalmente per gli oggetti eliminati dall'archivio S3 Glacier Instant Retrieval prima del termine minimo di 90 giorni.
region-EarlyDelete-GIR-SmObjects	GB/ora	Orario	Utilizzo dell'archiviazione ripartita proporzionalmente per gli oggetti di piccole dimensioni (inferiori a 128 KB) eliminati da S3 Glacier Instant Retrieval prima del termine minimo di 90 giorni.
region-EarlyDelete-SIA	GB/ora	Orario	Utilizzo dello storage ripartito proporzionalmente per gli oggetti eliminati da S3 Standard-IA prima del termine minimo di 30 giorni[3]
region-EarlyDelete-SIA-SmObjects	GB/ora	Orario	Utilizzo dello storage ripartito proporzionalmente per gli oggetti di piccole dimensioni (inferiori a 128 KB) eliminati da S3 Standard-IA prima del termine minimo di 30 giorni[3]

Tipo di utilizzo	unità	Granularità	Descrizione
region-EarlyDelete-ZIA	GB/ora	Orario	Utilizzo dello storage ripartito proporzionalmente per gli oggetti eliminati da S3 One Zone-IA prima del termine minimo di 30 giorni[3]
region-EarlyDelete-ZIA-SmObjects	GB/ora	Orario	Utilizzo dello storage ripartito proporzionalmente per gli oggetti di piccole dimensioni (inferiori a 128 KB) eliminati da S3 One Zone-IA prima del termine minimo di 30 giorni[3]
region-Expedited-Retrieval-Bytes	GB	Orario	La quantità di dati recuperati con le richieste S3 Glacier Flexible Retrieval di tipo Expedited
region-Inventory-Objects Listed	Oggetti	Orario	Numero di oggetti elencati per un gruppo di oggetti (gli oggetti sono raggruppati in base al bucket o al prefisso) con un elenco di inventario
region-Monitoring-Automation-INT	Oggetti	Orario	Numero di oggetti univoci monitorati e con livello assegnato automaticamente nella classe di storage S3 Intelligent-Tiering

Tipo di utilizzo	unità	Granularità	Descrizione
region-MRAP-Out-Bytes	GB	Orario	La quantità di dati trasferiti tramite un endpoint S3 Multi-Region Access Points dai bucket di una regione (prezzo di routing dei dati MRAP).
region-MRAP-In-Bytes	GB	Orario	La quantità di dati trasferiti tramite un endpoint S3 Multi-Region Access Points dai bucket di una regione (prezzi di routing dei dati MRAP).
regiongroup1-regiongroup2--MRAP-Out-Bytes	GB	Orario	La quantità di dati trasferiti tramite un endpoint S3 Multi-Region Access Points da un bucket in *regiongroup1* a un client situato all'esterno della rete. *regiongroup2* AWS
regiongroup1-regiongroup2--MRAP-In-Bytes	GB	Orario	La quantità di dati trasferiti tramite un endpoint S3 Multi-Region Access Points a un bucket in *regiongroup1* da un client situato all'esterno della rete. *regiongroup2* AWS

Tipo di utilizzo	unità	Granularità	Descrizione
region-OverwriteBytes-Copy-GDA	GB	Mensile	La quantità di dati sovrascritti da un'CopyObject operazione dallo storage S3 Glacier Deep Archive
region-OverwriteBytes-Copy-GIR	GB	Mensile	La quantità di dati sovrascritti da un'CopyObject operazione dallo storage S3 Glacier Instant Retrieval.
region-OverwriteBytes-Copy-GLACIER	GB	Mensile	La quantità di dati sovrascritti da un'operazione dallo storage S3 Glacier Flexible CopyObject Retrieval
region-OverwriteBytes-Copy-INT	GB	Mensile	La quantità di dati sovrascritti da un'operazione dallo storage S3 Intelligent-Tiering CopyObject
region-OverwriteBytes-Copy-RRS	GB	Mensile	La quantità di dati sovrascritti da un'CopyObject operazione dallo storage RRRS (Reduced Redundancy Storage)

Tipo di utilizzo	unità	Granularità	Descrizione
region-OverwriteBytes-Copy-SIA	GB	Mensile	La quantità di dati sovrascritti da un'CopyObject operazione dallo storage S3 Standard-IA
region-OverwriteBytes-Copy-STANDARD	GB	Mensile	La quantità di dati sovrascritti da un'CopyObject operazione dallo storage S3 Standard
region-OverwriteBytes-Copy-ZIA	GB	Mensile	La quantità di dati sovrascritti da un'CopyObject operazione dallo storage S3 One Zone-IA
region-OverwriteBytes-Put-GDA	GB	Mensile	La quantità di dati sovrascritti da un'PutObject operazione dallo storage S3 Glacier Deep Archive
region-OverwriteBytes-Put-GIR	GB	Mensile	La quantità di dati sovrascritti da un'PutObject operazione dallo storage S3 Glacier Instant Retrieval.
region-OverwriteBytes-Put-GLACIER	GB	Mensile	La quantità di dati sovrascritti da un'operazione dallo storage S3 Glacier Flexible PutObject Retrieval

Tipo di utilizzo	unità	Granularità	Descrizione
region-OverwriteBytes-Put-INT	GB	Mensile	La quantità di dati sovrascritti da un'operazione dallo storage S3 Intelligent-Tiering `PutObject`
region-OverwriteBytes-Put-RRS	GB	Mensile	La quantità di dati sovrascritti da un'`PutObject` operazione dallo storage RRRS (Reduced Redundancy Storage)
region-OverwriteBytes-Put-SIA	GB	Mensile	La quantità di dati sovrascritti da un'`PutObject` operazione dallo storage S3 Standard-IA
region-OverwriteBytes-Put-STANDARD	GB	Mensile	La quantità di dati sovrascritti da un'`PutObject` operazione dallo storage S3 Standard
region-OverwriteBytes-Put-ZIA	GB	Mensile	La quantità di dati sovrascritti da un'`PutObject` operazione dallo storage S3 One Zone-IA

Tipo di utilizzo	unità	Granularità	Descrizione
region1-*region2*-S3RTC-In-Bytes	GB	Mensile	La quantità di dati trasferiti per S3 Replication Time Control (S3 RTC) da *region2* a *region1* da,,, e operazioni PutObjectReplTime GetObjectReplTime InitiateMultipartUploadReplTime UploadPartReplTime CompleteMultipartUploadReplTime WriteACLReplTime
region1-*region2*-S3RTC-Out-Bytes	GB	Mensile	La quantità di dati trasferiti per S3 Replication Time Control (S3 RTC) da *region1* a *region2* da,,, e operazioni PutObjectReplTime GetObjectReplTime InitiateMultipartUploadReplTime UploadPartReplTime CompleteMultipartUploadReplTime WriteACLReplTime

Tipo di utilizzo	unità	Granularità	Descrizione
region-Requests-GDA-Tier1	Conteggio	Orario	Il numero diPUT,, COPY POST CreateMultipartUpl oad UploadPart , o CompleteMultipartUpload richieste sugli oggetti S3 Glacier Deep Archive 6
region-Requests-GDA-Tier2	Conteggio	Orario	Il numero GET e le HEAD richieste sugli oggetti S3 Glacier Deep Archive
region-Requests-GDA-Tier3	Conteggio	Orario	Numero di richieste di ripristino S3 Glacier Deep Archive di tipo standard
region-Requests-GDA-Tier5	Conteggio	Orario	Numero di richieste di ripristino S3 Glacier Deep Archive di tipo Bulk
region-Requests-GIR-Tier1	Conteggio	Orario	Il numero di o POST richieste sugli PUT oggetti COPY S3 Glacier Instant Retrieval.
region-Requests-GIR-Tier2	Conteggio	Orario	Il numero GET e tutte le altre richieste non S3 Glacier Instant Retrieval -Tier1 sugli oggetti S3 Glacier Instant Retrieval.

Tipo di utilizzo	unità	Granularità	Descrizione
region-Requests-GLACIER-Tier1	Conteggio	Orario	Il numero di,,, o richieste sugli oggetti S3 Glacier Flexible PUT Retrieval 6 COPY POST CreateMultipartUpload UploadPart CompleteMultipartUpload
region-Requests-GLACIER-Tier2	Conteggio	Orario	Il numero di richieste GET e tutte le altre richieste non elencate negli oggetti S3 Glacier Flexible Retrieval
region-Requests-INT-Tier1	Conteggio	Orario	Il numero di o POST richieste sugli oggetti S3 PUT COPY Intelligent-Tiering
region-Requests-INT-Tier2	Conteggio	Orario	Il numero GET e tutte le altre richieste non di livello 1 per oggetti S3 Intelligent-Tiering
region-Requests-SIA-Tier1	Conteggio	Orario	Il numero di o richieste sugli oggetti S3 Standard-IA PUT COPY POST
region-Requests-SIA-Tier2	Conteggio	Orario	Il numero GET e tutte le altre richieste non S3 Glacier Instant Retrieval -Tier1 sugli oggetti S3 Standard-IA

Tipo di utilizzo	unità	Granularità	Descrizione
region-Requests-Tier1	Conteggio	Orario	Il numero di o POST richieste per S3 Standard PUTCOPY, RRS e tag, più le richieste per tutti i bucket e gli oggetti LIST
region-Requests-Tier2	Conteggio	Orario	Il numero GET e tutte le altre richieste non di livello 1
region-Requests-Tier3	Conteggio	Orario	Numero totale di richieste del ciclo di vita verso S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive e richieste di ripristino standard di S3 Glacier Flexible Retrieval
region-Requests-Tier4	Conteggio	Orario	Numero di transizioni del ciclo di vita all'archivio S3 Glacier Instant Retrieval , S3 Intelligent-Tiering, S3 Standard-IA o S3 One Zone-IA
region-Requests-Tier5	Conteggio	Orario	Numero di richieste di ripristino S3 Glacier Flexible Retrieval di tipo Bulk
region-Requests-Tier6	Conteggio	Orario	Numero di richieste di ripristino S3 Glacier Flexible Retrieval di tipo Expedited

Tipo di utilizzo	unità	Granularità	Descrizione
region-Requests-Tier8	Conteggio	Orario	Il numero di richieste S3 Access Grants
region-Requests-XZ-Tier1	Conteggio	Orario	Il numero di COPY richieste PUT o sugli oggetti S3 Express One Zone
region-Requests-XZ-Tier2	Conteggio	Orario	Il numero di GET e tutte le altre richieste non S3 Express One Zone-Tier1 sugli oggetti S3 Express One Zone
region-Requests-ZIA-Tier1	Conteggio	Orario	Il numero di o richieste sugli oggetti S3 PUT One COPY Zone-IA POST
region-Requests-ZIA-Tier2	Conteggio	Orario	Il numero GET e tutte le altre richieste non S3 One Zone-IA-Tier1 sugli oggetti S3 One Zone-IA
region-Retrieval-GIR	GB	Orario	La quantità di dati recuperati dall'archivio S3 Glacier Instant Retrieval.
region-Retrieval-SIA	GB	Orario	La quantità di dati recuperati dallo storage S3 Standard-IA

Tipo di utilizzo	unità	Granularità	Descrizione
region-Retrieval-XZ	GB	Orario	La parte di dati che supera i 512 KB in una determinata richiesta di recupero (o) con storage S3 Express One Zone PUT COPY
region-Retrieval-ZIA	GB	Orario	La quantità di dati recuperati dallo storage S3 One Zone-IA
region-S3DSSE-In-Bytes	GB	Mensile	La quantità di dati con doppia crittografia da Amazon S3
region-S3DSSE-Out-Bytes	GB	Mensile	La quantità di dati con doppia crittografia decrittografati da Amazon S3
region-S3G-DataTransfer-In-Bytes	GB	Orario	La quantità di dati trasferiti ad Amazon S3 per il ripristino degli oggetti dall'archivio S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive
region-S3G-DataTransfer-Out-Bytes	GB	Orario	La quantità di dati trasferiti da Amazon S3 per la transizione degli oggetti all'archivio S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive

Tipo di utilizzo	unità	Granularità	Descrizione
region-Select-Returned-Bytes	GB	Orario	La quantità di dati restituiti con richieste Seleziona dallo storage S3 Standard
region-Select-Returned-GIR-Bytes	GB	Orario	La quantità di dati restituiti dall'archivio S3 Glacier Instant Retrieval con richieste Select.
region-Select-Returned-INT-Bytes	GB	Orario	La quantità di dati restituiti con richieste Seleziona dallo storage S3 Intelligent-Tiering
region-Select-Returned-SIA-Bytes	GB	Orario	La quantità di dati restituiti con richieste Seleziona dallo storage S3 Standard-IA
region-Select-Returned-ZIA-Bytes	GB	Orario	La quantità di dati restituiti con richieste Seleziona dallo storage S3 One Zone-IA
region-Select-Scanned-Bytes	GB	Orario	La quantità di dati scansionati con richieste Seleziona dallo storage S3 Standard
region-Select-Scanned-GIR-Bytes	GB	Orario	La quantità di dati scansionati dall'archivio S3 Glacier Instant Retrieval con richieste Select.

Tipo di utilizzo	unità	Granularità	Descrizione
region-Select-Scanned-INT-Bytes	GB	Orario	La quantità di dati scansionati con richieste Seleziona dallo storage S3 Intelligent-Tiering
region-Select-Scanned-SIA-Bytes	GB	Orario	La quantità di dati scansionati con richieste Seleziona dallo storage S3 Standard-IA
region-Select-Scanned-ZIA-Bytes	GB	Orario	La quantità di dati scansionati con richieste Seleziona dallo storage S3 One Zone-IA
region-Standard-Retrieval-Bytes	GB	Orario	La quantità di dati recuperati con richieste standard S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive
region-StorageAnalytics-ObjCount	Oggetti	Orario	Il numero di oggetti univoci monitorati in ciascuna configura zione di Storage Class Analysis.
region-StorageLens-ObjCount	Oggetti	Giornaliero	Il numero di oggetti univoci in ogni pannello di controllo di S3 Storage Lens monitorati da parametri e raccomand azioni avanzati di S3 Storage Lens.

Tipo di utilizzo	unità	Granularità	Descrizione
region-StorageLensFreeTier-ObjCount	Oggetti	Giornaliero	Il numero di oggetti univoci in ogni pannello di controllo di S3 Storage Lens monitorati da parametri di utilizzo di S3 Storage Lens.
StorageObjectCount	Conteggio	Giornaliero	Numero di oggetti archiviati in un determina to bucket
region-TagStorage-TagHrs	Tag-ora	Giornaliero	Tag complessivi su tutti gli oggetti del bucket indicati su base oraria
region-TimedStorage-ByteHrs	GB al mese	Giornaliero	Il numero di GB al mese in cui i dati sono stati archiviati nello storage S3 Standard
region-TimedStorage-GDA-ByteHrs	GB al mese	Giornaliero	Il numero di GB al mese in cui i dati sono stati archiviati nello storage S3 Glacier Deep Archive
region-TimedStorage-GDA-Staging	GB al mese	Giornaliero	Il numero di GB al mese in cui i dati sono stati archiviati nello storage di staging S3 Glacier Deep Archive
region-TimedStorage-GIR-ByteHrs	GB al mese	Giornaliero	Il numero di GB al mese in cui i dati sono stati archiviati nello storage S3 Glacier Instant Retrieval.

Tipo di utilizzo	unità	Granularità	Descrizione
region-TimedStorage-GIR-SmObjects	GB al mese	Giornaliero	Il numero di GB al mese in cui piccoli oggetti (inferiori a 128 KB) sono stati archiviati nello storage S3 Glacier Instant Retrieval.
region-TimedStorage-GlacierByteHrs	GB al mese	Giornaliero	Il numero di GB al mese in cui i dati sono stati archiviati nello storage S3 Glacier Flexible Retrieval
region-TimedStorage-GlacierStaging	GB al mese	Giornaliero	Il numero di GB al mese in cui i dati sono stati archiviati nello storage di staging S3 Glacier Flexible Retrieval
region-TimedStorage-INT-FA-ByteHrs	GB al mese	Giornaliero	Il numero di GB al mese in cui i dati sono stati archiviati nel livello Frequent Access dello storage S3 Intelligent-Tiering 5
region-TimedStorage-INT-IA-ByteHrs	GB al mese	Giornaliero	Il numero di GB al mese in cui i dati sono stati archiviati nel livello Infrequent Access dello storage S3 Intelligent-Tiering
region-TimedStorage-INT-AA-ByteHrs	GB al mese	Giornaliero	Il numero di GB al mese in cui i dati sono stati archiviati nel livello Archive Access dello storage S3 Intelligent-Tiering

Tipo di utilizzo	unità	Granularità	Descrizione
region-TimedStorage-INT-AIA-ByteHrs	GB al mese	Giornaliero	Il numero di GB al mese in cui i dati sono stati archiviati nel livello Archive Instant Access dello storage S3 Intelligent-Tiering
region-TimedStorage-INT-DAA-ByteHrs	GB al mese	Giornaliero	Il numero di GB al mese in cui i dati sono stati archiviati nel livello Deep Archive Access dello storage S3 Intelligent-Tiering
region-TimedStorage-RRS-ByteHrs	GB al mese	Giornaliero	Il numero di GB al mese in cui i dati sono stati archiviati nello storage RRRS (Reduced Redundancy Storage)
region-TimedStorage-SIA-ByteHrs	GB al mese	Giornaliero	Il numero di GB al mese in cui i dati sono stati archiviati nello storage S3 Standard-IA
region-TimedStorage-SIA-SmObjects	GB al mese	Giornaliero	Il numero di GB al mese in cui oggetti di piccole dimensioni (inferiori a 128 KB) sono stati archiviati nello storage S3 Standard-IA 4
region-TimedStorage-XZ-ByteHrs	GB al mese	Giornaliero	Il numero di GB al mese in cui i dati sono stati archiviati nello storage S3 Express One Zone

Tipo di utilizzo	unità	Granularità	Descrizione
region-TimedStorage-ZIA-ByteHrs	GB al mese	Giornaliero	Il numero di GB al mese in cui i dati sono stati archiviati nello storage S3 One Zone-IA
region-TimedStorage-ZIA-SmObjects	GB al mese	Giornaliero	Il numero di GB al mese in cui piccoli oggetti (inferiori a 128 KB) sono stati archiviati nello storage S3 One Zone-IA
region-Upload-XZ	GB	Orario	La quantità di dati che supera i 512 KB in una determinata richiesta di caricamento (PUToCOPY) con S3 Express One Zone

Note

1. Se termini un trasferimento prima del completamento, la quantità di dati trasferiti può superare la quantità di dati ricevuti dall'applicazione. Questa discrepanza può verificarsi perché una richiesta di interruzione del trasferimento non può essere eseguita istantaneamente e una certa quantità di dati potrebbe essere in transito, in attesa dell'esecuzione della richiesta di terminazione. Questi dati in transito vengono fatturati come dati trasferiti «in uscita».
2. Quando gli oggetti archiviati nella classe di storage S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive vengono eliminati, sovrascritti o trasferiti a una classe di storage diversa prima che sia trascorso l'impegno di storage minimo, ovvero 90 giorni per S3 Glacier Instant Retrieval e S3 Glacier Flexible Retrieval, o 180 giorni per S3 Glacier Deep Archive, è previsto un addebito ripartito proporzionalmente per gigabyte per i giorni rimanenti.
3. Per gli oggetti che si trovano nello storage S3 Standard-IA o S3 One Zone-IA, quando vengono eliminati, sovrascritti o trasferiti a una classe di archiviazione diversa prima di 30 giorni, viene applicato un addebito ripartito proporzionalmente per gigabyte per i giorni rimanenti.

4. Per gli oggetti di piccole dimensioni (inferiori a 128 KB) che si trovano nello storage S3 Standard-IA o S3 One Zone-IA, quando vengono eliminati, sovrascritti o trasferiti a una classe di storage diversa prima di 30 giorni, viene applicato un addebito ripartito proporzionalmente per gigabyte per i giorni rimanenti.
5. Non sono previste dimensioni fatturabili minime per gli oggetti nella classe di archiviazione S3 Intelligent-Tiering. Gli oggetti di dimensioni inferiori a 128 KB non sono monitorati né idonei per il tiering automatico. Gli oggetti più piccoli vengono archiviati nel livello Accesso frequente della classe S3 Intelligent-Tiering.
6. Quando avvii una `CreateMultipartUpload` o più `UploadPartCopy` richieste alle classi di storage S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, le richieste vengono fatturate alle tariffe di richiesta S3 Standard fino al completamento del caricamento multiparte. `UploadPart` Una volta completato il caricamento, la singola `CompleteMultipartUpload` richiesta viene fatturata alla tariffa dello storage S3 Glacier di destinazione. `PUT` Le parti di caricamento multiparte in corso per `PUT` a sulla classe di storage S3 Glacier Flexible Retrieval vengono fatturate come S3 Glacier Flexible Retrieval Staging Storage alle tariffe di archiviazione S3 Standard fino al completamento del caricamento. Analogamente, le parti di caricamento multiparte in corso `PUT` per una classe di storage S3 Glacier Deep Archive vengono fatturate come S3 Glacier Deep Archive Staging Storage alle tariffe di storage S3 Standard fino al completamento del caricamento.
7. S3 Express One Zone applica una tariffa fissa per richiesta per richieste di dimensioni fino a 512 KB. Viene applicato un costo aggiuntivo per GB per `PUT` le richieste e le `GET` richieste per la parte di richiesta superiore a 512 KB.
8. Per informazioni sulle funzionalità supportate per la classe di archiviazione S3 Express One Zone, consulta Funzionalità Amazon S3 non supportate da S3 Express One Zone.
9. I tipi di utilizzo con unità fatturate in GB vengono calcolati in byte nei report sull'utilizzo.
10. Un GB al mese si ricava prendendo il numero totale di GB all'ora, aggregandoli nel corso di un mese e quindi dividendolo per il numero di ore di quel mese. Per ulteriori informazioni, consulta Domande frequenti: Come verrà addebitato e fatturato l'utilizzo di Amazon S3?

> **Note**
>
> In generale, ai proprietari di bucket S3 vengono fatturate le richieste con risposte HTTP `200 OK` riuscite e risposte di errore del client HTTP. `4XX` Ai proprietari di bucket non vengono addebitate le risposte agli errori `5XX` del server HTTP, come gli errori HTTP. `503 Slow Down` Per ulteriori informazioni sui codici di errore S3 in HTTP `3XX` e sui codici di `4XX` stato non fatturati, consulta. Risposte agli errori di fatturazione per Amazon S3 Per ulteriori

informazioni sulla fatturazione degli addebiti se il tuo bucket è configurato come bucket Requester Pays, consulta. Come funzionano i pagamenti a carico del richiedente

Operazioni di monitoraggio nei report di utilizzo

Le operazioni descrivono l'azione intrapresa sull' AWS oggetto o sul bucket in base al tipo di utilizzo specificato. Le operazioni sono indicate con codici autoesplicativi, ad esempio `PutObject` o `ListBucket`. Fare riferimento a tali codici per vedere quali operazioni nel bucket hanno generato un tipo di utilizzo specifico. Quando crei un report di utilizzo, puoi scegliere di specificare All Operations (Tutte le operazioni) o un'operazione specifica, ad esempio `GetObject`.

Ulteriori informazioni

- AWS rapporto di utilizzo per Amazon S3
- AWS Billing report per Amazon S3
- Prezzi di Amazon S3
- Domande frequenti su Amazon S3

Risposte agli errori di fatturazione per Amazon S3

Important

Il 13 maggio 2024, abbiamo iniziato a implementare una modifica per eliminare gli addebiti per le richieste non autorizzate non avviate dal proprietario del bucket. Una volta completata l'implementazione di questa modifica, i proprietari dei bucket non dovranno mai sostenere costi di richiesta o larghezza di banda per le richieste che restituiscono errori `AccessDenied` (HTTP`403 Forbidden`) quando tali richieste vengono avviate dall'esterno del loro account o organizzazione individuale. AWS AWS La pagina corrente mostra un elenco completo dei codici HTTP `3XX` e di `4XX` stato che non verranno fatturati. Questa modifica alla fatturazione non richiede aggiornamenti delle applicazioni e si applica a tutti i bucket S3. Una volta completata l'implementazione di questa modifica Regioni AWS, aggiorneremo la nostra documentazione.

In generale, ai proprietari di bucket S3 vengono fatturate le richieste con risposte HTTP `200 OK` riuscite e risposte di errore `4XX` del client HTTP. Ai proprietari di bucket non vengono addebitate

le risposte agli errori 5XX del server HTTP, come gli errori HTTP. 503 Slow Down Per ulteriori informazioni sugli addebiti di fatturazione se il bucket è configurato come bucket Requester Pays, consulta. Come funzionano i pagamenti a carico del richiedente

La tabella seguente elenca i codici di errore specifici in HTTP 3XX e i codici di 4XX stato che non vengono fatturati. Per i bucket configurati con l'hosting di siti Web, le richieste applicabili e gli altri costi verranno comunque applicati quando S3 restituisce un documento di errore personalizzato o per reindirizzamenti personalizzati.

> **Note**
>
> Per AccessDenied (HTTP403 Forbidden), S3 non addebita alcun costo al proprietario del bucket quando la richiesta viene avviata al di fuori dell'account individuale AWS del proprietario del bucket o dell'organizzazione del proprietario del bucket. AWS

Codice di stato HTTP	Codice di errore	Descrizione del codice di errore	
301 Moved Permanently (301 Spostato definitivamente)	PermanentRedirect	Il bucket a cui si sta tentando di accedere deve essere indirizzato utilizzando l'endpoint specificato. Invia tutte le richieste future a questo endpoint.	
	PermanentRedirectControlError	L'operazione API a cui stai tentando di accedere deve essere gestita utilizzando l'endpoint specificato. Invia tutte le richieste future a questo endpoint.	
307 Reindirizzamento	TemporaryRedirect	Verrai reindirizzato al bucket durante l'aggiornamento del server	

Codice di stato HTTP	Codice di errore	Descrizione del codice di errore
temporaneo		Domain Name System (DNS).
400 Richiesta non valida	AuthorizationHeaderMalformed	L'intestazione di autorizzazione che hai fornito non è valida.
	AuthorizationQueryParametersError	I parametri della richiesta di autorizzazione che hai fornito non sono validi.
	ExpiredToken	Il token fornito è scaduto.
	IllegalLocationConstraintException	Stai tentando di accedere a un bucket da una regione diversa da quella in cui esiste il bucket. Per evitare questo errore, usa l'`--region`opzione. Ad esempio: `aws s3 cp` *`awsexample.txt`* `s3://` *`example-s3-bucket`*`/ --region` *`ap-east-1`* .

Codice di stato HTTP	Codice di errore	Descrizione del codice di errore	
	InvalidArgument	Questo errore può verificarsi per i seguenti motivi: • L'argomento specificato non era valido. • Nella richiesta mancava un'intest azione obbligatoria. • L'argomento specifica to era incompleto o nel formato errato. • L'argomento specifica to deve avere una lunghezza maggiore o uguale a 3.	
	InvalidDigest	Il valore di content-MD5 o checksum specificato non è valido.	
	InvalidEncryptionA lgorithmError	La richiesta di crittografia specificata non è valida. Il valore valido è AES256.	

Codice di stato HTTP	Codice di errore	Descrizione del codice di errore	
	InvalidRequest	Questo errore può verificarsi per i seguenti motivi: • La richiesta utilizza la versione della firma errata. Usa `AWS4-HMAC-SHA256` (Signature Version 4). • Un punto di accesso può essere creato solo per un bucket esistente . • Il punto di accesso non si trova in uno stato in cui può essere eliminato. • Un punto di accesso può essere elencato solo per un bucket esistente. • Il token successivo non è valido. • È necessario specifica re almeno un'azione in una regola del ciclo di vita. •	

Codice di stato HTTP	Codice di errore	Descrizione del codice di errore	
		È necessario specifica re almeno una regola del ciclo di vita. • Il numero di regole del ciclo di vita non deve superare il limite consentito di 1000 regole. • L'intervallo per il `MaxResults` parametro non è valido. • Le richieste SOAP devono essere effettuate tramite una connessione HTTPS. • Amazon S3 Transfer Acceleration non è supportato per i bucket con nomi non conformi al DNS. • Amazon S3 Transfer Acceleration non è supportato per i bucket con punti (.) nel nome. • L'endpoint Amazon S3 Transfer Acceleration supporta solo richieste in stile virtuale.	

Codice di stato HTTP	Codice di errore	Descrizione del codice di errore	
		• Amazon S3 Transfer Acceleration non è configurato su questo bucket. • Amazon S3 Transfer Acceleration è disabilitato su questo bucket. • Amazon S3 Transfer Acceleration non è supportato in questo bucket. Per assistenza, contatta. AWS Support • Amazon S3 Transfer Acceleration non può essere abilitato su questo bucket. Per assistenza, contatta. AWS Support • Valori in conflitto forniti nelle intestazioni HTTP e nei parametri di query. • Valori in conflitto forniti nelle intestazioni HTTP e nei campi del modulo POST. • CopyObject richiesta effettuata su oggetti di	

Codice di stato HTTP	Codice di errore	Descrizione del codice di errore	
		dimensioni superiori a 5 GB.	
	Richiesta SOAP non valida	Il corpo della richiesta SOAP non è valido.	
	InvalidStorageClass	La classe di archiviazione specificata non è valida.	
	InvalidTag	La richiesta contiene un input di tag non valido. Ad esempio, la richiesta potrebbe contenere chiavi, chiavi o valori duplicati troppo lunghi o tag di sistema.	
	InvalidToken	Il token fornito non è valido o altrimenti non è valido.	
	URI non valido	L'URI specificato non può essere analizzato.	
	KeyTooLongError	La tua chiave è troppo lunga.	
	Errore ACL non valido	L'ACL che hai fornito non è stato formato correttamente o non è stato convalidato rispetto allo schema pubblicato.	

Codice di stato HTTP	Codice di errore	Descrizione del codice di errore	
	PostRequest non valido	Il corpo della richiesta POST non è costituito da dati multiparti/moduli ben formati.	
	XML non valido	Il codice XML fornito non è ben formato o non è stato convalidato rispetto allo schema pubblicato.	
	MaxPostPreDataLengthExceededError	I campi di richiesta POST precedenti al file di caricamento erano troppo grandi.	
	MetadataTooLarge	Le intestazioni dei metadati superano la dimensione massima consentita per i metadati.	
	MissingRequestBodyError	Hai inviato un documento XML vuoto come richiesta.	
	MissingSecurityHeader	Nella tua richiesta manca un'intestazione obbligatoria.	
	NoLoggingStatusForKey	Non esiste una sottorisorsa sullo stato della registrazione di una chiave.	

Codice di stato HTTP	Codice di errore	Descrizione del codice di errore	
	RequestHeaderSectionTooLarge	L'intestazione della richiesta e i parametri di query utilizzati per far sì che la richiesta superi le dimensioni massime consentite	
	UnexpectedContent	Questa richiesta contiene contenuti non supportati.	
	UserKeyMustBeSpecified	La richiesta POST del bucket deve contenere il nome di campo specificato. Se è specificato, controlla l'ordine dei campi.	
	IncorrectEndpoint	Il bucket specificato esiste in un'altra regione. Indirizza le richieste all'endpoint corretto.	
403 Non consentito	RequestTimeTooSkewed	La differenza tra l'ora della richiesta e l'ora del server è troppo grande.	

Codice di stato HTTP	Codice di errore	Descrizione del codice di errore	
	SignatureDoesNotMatch	La firma di richiesta calcolata dal server non corrisponde alla firma fornita. Controlla la tua chiave di accesso AWS segreta e il metodo di firma. Per ulteriori informazioni, consulta Autenticazione REST e Autenticazione SOAP.	
	NotSignedUp	Il tuo account non è registrato per il servizio Amazon S3. È necessari o registrarsi prima di poter utilizzare Amazon S3. Puoi registrarti al seguente URL: https://a ws.amazon.com/s3	
	InvalidSecurity	Le credenziali di sicurezza fornite non sono valide.	
	InvalidPayer	Tutti gli accessi a questo oggetto sono stati disabilit ati. Per ulteriore assistenz a, consulta Contattaci.	
	InvalidAccessKeyId	L'ID della chiave di AWS accesso che hai fornito non esiste nei nostri archivi.	

Codice di stato HTTP	Codice di errore	Descrizione del codice di errore	
	AccountProblem	C'è un problema Account AWS che impedisce il corretto completam ento dell'operazione. Per ulteriore assistenza, consulta Contattaci.	
	UnauthorizedAccessError	Applicabile solo nelle regioni della Cina. Restituito quando viene effettuata una richiesta a un bucket che non dispone di una licenza ICP. Per ulteriori informazioni, vedere ICP Recordal.	
404 Not Found (404 Non trovato)	NoSuchUpload	Il caricamento multipart e specificato non esiste. L'ID di caricamen to potrebbe non essere valido oppure il caricamento in più parti potrebbe essere stato interrotto o completato.	
	NoSuchWebsiteConfi guration	Il bucket specificato non ha una configurazione del sito Web.	
Metodo 405 non consentito	MethodNotAllowed	Il metodo specificato non è consentito su questa risorsa.	

Codice di stato HTTP	Codice di errore	Descrizione del codice di errore	
409 Conflitto	BucketAlreadyExists	Il nome del bucket richiesto non è disponibile. Lo spazio dei nomi del bucket è condiviso da tutti gli utenti del sistema. Specificare un nome diverso e riprovare.	
	InvalidBucketState	La richiesta non è valida per lo stato corrente del bucket.	
	OperationAborted	È attualmente in corso un'operazione condizionale in conflitto su questa risorsa. Riprova.	
4.1.1 Lunghezza richiesta	MissingContentLength	È necessario fornire l'intestazione HTTP Content-Length.	
4.1.2 Precondizione non riuscita	RequestIsNotMultiPartContent	Una richiesta POST del bucket deve essere del tipo allegato multipart/form-data.	

Filtro e recupero dei dati tramite Amazon S3 Select

Con Amazon S3 Select è possibile utilizzare un linguaggio di query strutturata (SQL) per filtrare i contenuti di oggetti Amazon S3 e recuperare solo il sottoinsieme di dati necessario. Utilizzando Amazon S3 Select per filtrare questi dati, è possibile ridurre la quantità di dati trasferiti da Amazon S3, riducendo il costo e la latenza di recupero dei dati.

Amazon S3 Select consente di eseguire query su un solo oggetto alla volta. Funziona su un oggetto archiviato in formato CSV, JSON o. Apache Parquet Funziona anche con un oggetto compresso con GZIP o BZIP2 (solo per oggetti CSV e JSON) e un oggetto crittografato lato server. Puoi specificare il formato dei risultati come CSV o JSON e determinare la modalità di delimitazione dei record nel risultato.

Passi espressioni SQL ad Amazon S3 nella richiesta. Amazon S3 Select supporta un sottoinsieme di SQL. Per ulteriori informazioni sugli elementi SQL supportati da Amazon S3 Select, consulta Documentazione di riferimento su SQL per Amazon S3 Select.

Puoi eseguire query SQL utilizzando la console Amazon S3, il AWS CLI(), AWS Command Line Interface l'operazione API REST o `SelectObjectContent` AWS gli SDK.

> **Note**
>
> La console di Amazon S3 limita la quantità di dati restituiti a 40 MB. Per recuperare più dati, usa l'o l' AWS CLI API.

Requisiti e limiti

Di seguito vengono riportati i requisiti per l'utilizzo di Amazon S3 Select:

- Devi disporre delle autorizzazioni `s3:GetObject` per l'oggetto su esegui la query.
- Se l'oggetto su cui esegui la query è crittografato con una chiave di crittografia lato server fornita dal cliente (SSE-C), devi utilizzare `https` e fornire la chiave di crittografia nella richiesta.

Per l'utilizzo di Amazon S3 Select si applicano i seguenti limiti:

- S3 Select può interrogare solo un oggetto per richiesta.
- La lunghezza massima di un'espressione SQL è di 256 KB.
- La lunghezza massima di un record nell'input o nel risultato è di 1 MB.
- Amazon S3 Select può emettere solo dati nidificati utilizzando il formato di output JSON.
- Non è possibile interrogare un oggetto archiviato nelle classi di storage S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive o Reduced Redundancy Storage (RRS). Inoltre, non è possibile interrogare un oggetto archiviato nel livello S3 Intelligent-Tiering Archive Access o nel livello S3

Intelligent-Tiering Deep Archive Access. Per ulteriori informazioni sulle classi di storage, consulta Utilizzo delle classi di storage di Amazon S3.

Si applicano limitazioni aggiuntive quando si utilizza Amazon S3 Select con un Parquet oggetto:

- Amazon S3 Select supporta solo la compressione a colonne con GZIP o Snappy. Amazon S3 Select non supporta la compressione dell'intero oggetto per un oggetto. Parquet
- Amazon S3 Select non supporta l'output Parquet. È necessario specificare il formato di output, ad esempio CSV o JSON.
- La dimensione massima del gruppo di righe non compresso è 256 MB.
- È necessario utilizzare i tipi di dati specificati nello schema dell'oggetto.
- Se si seleziona un campo ripetuto, viene restituito solo l'ultimo valore.

Costruzione di una richiesta

Quando costruisci una richiesta, devi fornire i dettagli dell'oggetto su cui si sta eseguendo la query utilizzando un oggetto `InputSerialization`. Fornisci i dettagli del modo in cui i risultati vengono restituiti utilizzando un oggetto `OutputSerialization`. Includi anche l'espressione SQL utilizzata da Amazon S3 per filtrare la richiesta.

Per ulteriori informazioni sulla creazione di una richiesta Amazon S3 Select, consulta SelectObjectContent nella Documentazione di riferimento delle API di Amazon Simple Storage Service. Uno degli esempi di codice SDK è anche disponibile nelle seguenti sezioni.

Richieste che utilizzano intervalli di scansione

Amazon S3 Select consente di eseguire la scansione di un sottoinsieme di un oggetto specificando un intervallo di byte su cui eseguire la query. Ciò consente di parallelizzare la scansione dell'intero oggetto dividendo il lavoro tra due richieste Amazon S3 Select separate per una serie di intervalli di scansione senza sovrapposizione.

Gli intervalli di scansione non devono essere allineati con i limiti di record. Una richiesta Amazon S3 Select di intervallo di scansione viene eseguita nell'intervallo di byte specificato. Un record che inizia nell'intervallo di scansione specificato ma che si estende oltre verrà elaborato dalla query. Ad esempio, di seguito viene mostrato un oggetto Amazon S3 contenente una serie di record in formato CSV delimitato da righe:

```
A,B
C,D
D,E
E,F
G,H
I,J
```

Viene utilizzato il parametro `ScanRange` di Amazon S3 Select e Start a (Byte) 1 ed End a (Byte) 4. Pertanto, l'intervallo di scansione inizia a "`,`" e la scansione verrà eseguita fino alla fine del record che inizia a `C`. La richiesta dell'intervallo di scansione restituirà il risultato `C, D` perché questa è la fine del record.

L'intervallo di scansione di Amazon S3 Select richiede supportoParquet, file CSV (senza delimitatori tra virgolette) o oggetti JSON (solo in modalità). `LINES` Gli oggetti CSV e JSON non devono essere compressi. Per gli oggetti CSV e JSON basati su righe, quando viene specificato un intervallo di scansione come parte della richiesta Amazon S3 Select, vengono elaborati tutti i record che iniziano nell'intervallo di scansione. Per gli oggetti Parquet, vengono elaborati tutti i gruppi di righe che iniziano all'interno dell'intervallo di scansione richiesto.

Le richieste dell'intervallo di scansione di Amazon S3 Select possono essere utilizzate con l' AWS CLI API Amazon S3 e gli SDK. AWS Per questa caratteristica, è possibile utilizzare il parametro `ScanRange` nella richiesta Amazon S3 Select. Per ulteriori informazioni, consulta SelectObjectContent in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Errori

Amazon S3 Select restituisce un codice di errore e un messaggio di errore associato quando si verifica un problema durante il tentativo di esecuzione di una query. Per un elenco di codici di errore e descrizioni, consulta la sezione relativa all'elenco dei codici di errore relativo al contenuto dell'oggetto SELECT nella pagina delle risposte agli errori nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Per maggiori informazioni su Amazon S3 Select, consulta gli argomenti seguenti:

Argomenti

- Esempi di utilizzo di Amazon S3 Select su un oggetto
- Documentazione di riferimento su SQL per Amazon S3 Select

Esempi di utilizzo di Amazon S3 Select su un oggetto

Puoi utilizzare S3 Select per selezionare il contenuto da un oggetto utilizzando la console Amazon S3, l'API REST e AWS gli SDK.

Per ulteriori informazioni sulle funzioni SQL supportate per S3 Select, consulta Funzioni SQL.

Utilizzo della console S3

Per selezionare il contenuto da un oggetto nella console Amazon S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Scegli il bucket che contiene l'oggetto da cui desideri selezionare il contenuto, quindi scegli il nome dell'oggetto.
4. Scegli Azioni oggetto e Interroga con S3 Select.
5. Configura impostazioni di input, in base al formato dei dati di input.
6. Configura impostazioni di output, in base al formato dell'output che desideri ricevere.
7. Per estrarre i record dall'oggetto scelto, in query SQL, inserisci i comandi SELECT SQL. Per ulteriori informazioni su come scrivere comandi SQL, consulta Documentazione di riferimento su SQL per Amazon S3 Select.
8. Dopo aver inserito le query SQL, scegli Esegui query SQL. Quindi, in Risultati della query, puoi visualizzare i risultati delle tue query SQL.

Utilizzo di REST API

Puoi utilizzare gli AWS SDK per selezionare il contenuto da un oggetto. Tuttavia, se l'applicazione lo richiede, è possibile inviare richieste REST direttamente. Per ulteriori informazioni sul formato della richiesta e della risposta, consulta SelectObjectContent.

Utilizzo degli SDK AWS

Puoi usare Amazon S3 Select per selezionare parte del contenuto di un oggetto utilizzando il `selectObjectContent` metodo. Se questo metodo ha esito positivo, restituisce i risultati dell'espressione SQL.

Java

Il codice Java seguente restituisce il valore della prima colonna per ogni record archiviato in un oggetto contenente dati archiviati in formato CSV. Richiede anche che vengano restituiti messaggi `Progress` e `Stats`. Fornire un nome bucket e un oggetto validi contenenti dati in formato CSV.

Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
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

Per un JavaScript esempio che utilizza l'operazione AWS SDK for JavaScript con l'`SelectObjectContent`API S3 per selezionare i record dai file JSON e CSV archiviati in Amazon S3, consulta il post sul blog Introduzione al supporto per Amazon S3 Select nel. AWS SDK for JavaScript

Python

Per un esempio Python sull'utilizzo delle query SQL per cercare i dati caricati su Amazon S3 come file CSV (Comma-Separated Value) utilizzando S3 Select, vedere il post del blog Interrogazione di dati senza server o database tramite Amazon S3 Select.

Documentazione di riferimento su SQL per Amazon S3 Select

Questa documentazione di riferimento contiene una descrizione degli elementi SQL supportati da Amazon S3 Select.

Argomenti

- SELECT command
- Tipi di dati
- Operatori
- Parole chiave riservate
- Funzioni SQL

SELECT command

Amazon S3 Select supporta soltanto il comando SQL `SELECT`. Le seguenti clausole standard ANSI sono supportate per `SELECT`:

- `SELECT` Elenco
- `FROM`Clausola
- `WHERE`Clausola
- `LIMIT`Clausola

> **Note**
>
> Al momento, le query di Amazon S3 Select non supportano query secondarie o join.

SELECT Elenco

L'elenco SELECT assegna un nome a colonne, funzioni ed espressioni che la query deve restituire. L'elenco rappresenta l'output della query.

```
SELECT *
SELECT projection1 AS column_alias_1, projection2 AS column_alias_2
```

La prima forma di SELECT con * (asterisco) restituisce ogni riga che ha passato la clausola , così com'è. Il secondo formato di SELECT crea una riga con la proiezione delle espressioni scalari di output definite dall'utente ***projection1*** e ***projection2*** per ogni colonna.

FROMClausola

Amazon S3 Select supporta i seguenti formati della clausola FROM:

```
FROM table_name
FROM table_name alias
FROM table_name AS alias
```

In ogni forma della clausola FROM, table_name è il S3Object che viene interrogato. Gli utenti abituati a database relazionali tradizionali possono considerare questo come uno schema di database che contiene più viste di una tabella.

Nel linguaggio SQL standard, la clausola FROM crea righe che vengono filtrate nella clausola WHERE e proiettate nell'elenco SELECT.

Per tutti gli oggetti JSON archiviati in Amazon S3 Select, puoi anche utilizzare i seguenti moduli della clausola FROM:

```
FROM S3Object[*].path
FROM S3Object[*].path alias
FROM S3Object[*].path AS alias
```

Utilizzando questo modulo della clausola FROM, puoi selezionare da array o oggetti inclusi in un oggetto JSON. Puoi specificare path mediante uno dei seguenti moduli:

- Per nome (in un oggetto): .*name* o ['*name*']
- Per indice (in un array): [*index*]
- Per carattere jolly (in un oggetto): .*

- Per carattere jolly (in un array): `[*]`

> **Note**
>
> - Questo modulo della clausola `FROM` è utilizzabile solo con oggetti JSON.
> - I caratteri jolly emettono sempre almeno un record. Se nessun record corrisponde, Amazon S3 Select emette il valore `MISSING`. Durante la serializzazione dell'output (dopo che la query è stata completata), Amazon S3 Select sostituisce i valori `MISSING` con record vuoti.
> - Le funzioni di aggregazione (`AVG`, `COUNT`, `MAX`, `MIN` e `SUM`) ignorano i valori `MISSING`.
> - Se non fornisci un alias quando utilizzi un carattere jolly, puoi fare riferimento alla riga utilizzando l'ultimo elemento nel percorso. Ad esempio, puoi selezionare tutti i prezzi da un elenco di libri utilizzando la query `SELECT price FROM S3Object[*].books[*].price`. Se il percorso termina con un carattere jolly anziché con un nome, puoi utilizzare il valore `_1` per fare riferimento alla riga. Ad esempio, anziché `SELECT price FROM S3Object[*].books[*].price`, puoi usare la query `SELECT _1.price FROM S3Object[*].books[*]`.
> - Amazon S3 Select considera sempre un documento JSON come un array di valori a livello di radice. Pertanto, anche se l'oggetto JSON che stai interrogando ha solo un elemento radice, la clausola `FROM` deve iniziare con `S3Object[*]`. Tuttavia, per motivi di compatibilità, Amazon S3 Select consente di omettere il carattere jolly se non si include un percorso. Pertanto, la clausola completa `FROM S3Object` è equivalente a `FROM S3Object[*] as S3Object`. Se includi un percorso, devi utilizzare anche il carattere jolly. Pertanto `FROM S3Object` e `FROM S3Object[*].`*`path`* sono entrambe clausole valide, ma `FROM S3Object.`*`path`* non è valida.

Example

Esempi:

Esempio 1

Questo esempio mostra i risultati utilizzando i seguenti set di dati e query:

```
{ "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 ciascun risultato per i seguenti motivi:

- `{"id":"id-1"}`— `S3Object[0].Rules[0].id` ha prodotto una corrispondenza.
- `{}` - `S3Object[0].Rules[1].id` non ha prodotto una corrispondenza con un record, quindi Amazon S3 Select ha emesso un valore `MISSING`, che è stato quindi modificato in un record vuoto durante la serializzazione dell'output e restituito.
- `{"id":"id-2"}`— `S3Object[0].Rules[2].id` ha prodotto una corrispondenza.
- `{}`-`S3Object[1]` non ha prodotto una corrispondenza in `Rules`, quindi Amazon S3 Select ha emesso un valore `MISSING`, che è stato quindi modificato in un record vuoto durante la serializzazione dell'output e restituito.

Se non desideri che Amazon S3 Select restituisca record vuoti quando non trova una corrispondenza, puoi testare il valore `MISSING`. La seguente query restituisce gli stessi risultati della query precedente, ma con i valori vuoti omessi:

```
SELECT id FROM S3Object[*].Rules[*].id WHERE id IS NOT MISSING
```

```
{"id":"1"}
{"id":"2"}
```

Esempio 2

Questo esempio mostra i risultati utilizzando i seguenti set di dati e query:

```
{ "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"}
```

WHEREClausola

La sintassi della clausola `WHERE` è la seguente:

```
WHERE condition
```

La clausola `WHERE` filtra le righe in base alla *`condition`*. Una condizione è un'espressione che genera un risultato booleano. Solo le righe per le quali la condizione è `TRUE` sono restituite nel risultato.

LIMITClausola

La sintassi della clausola `LIMIT` è la seguente:

```
LIMIT number
```

La clausola `LIMIT` limita il numero di record che desideri vengano restituiti dalla query in base al *`number`* specificato.

Accesso agli attributi

Le clausole `SELECT` e `WHERE` possono fare riferimento a record di dati utilizzando uno dei metodi descritti nelle seguenti sezioni, a seconda che il file su cui viene eseguita la query sia in formato CSV o JSON.

CSV

- Numeri di colonna: puoi fare riferimento alla N-esima colonna di una riga con il nome di colonna , dove N è la posizione della colonna. La numerazione delle posizioni inizia da 1. Ad esempio, la prima colonna è denominata _1 e la seconda è denominata _2.

  Puoi fare riferimento a una colonna con *_N* o *alias*._*N*. Ad esempio, _2 e myAlias._2 sono entrambi modi validi di fare riferimento a una colonna nell'elenco SELECT e nella clausola WHERE.
- Intestazioni di colonna: per gli oggetti in formato CSV che hanno una riga di intestazione, le intestazioni sono disponibili per l'elenco SELECT e la clausola WHERE. In particolare, come nel linguaggio SQL classico, all'interno di espressioni con le clausole SELECT e WHERE è possibile fare riferimento alle colonne in base a *alias*.*column_name* o *column_name*.

JSON

- Documento: puoi accedere ai campi di documento JSON con *alias*.*name*. È inoltre possibile accedere ai campi nidificati, ad esempio *alias*.*name1*.*name2*.*name3*.
- Elenco: puoi accedere agli elementi in un elenco JSON utilizzando indici a base zero con l'operatore []. Ad esempio, puoi accedere al secondo elemento di un elenco con *alias*[1]. È possibile abbinare l'accesso agli elementi dell'elenco con i campi, *alias*.*name1*.*name2*[1].*name3*, ad esempio.
- Esempi: considera questo oggetto JSON come un set di dati di esempio:

```
{"name": "Susan Smith",
"org": "engineering",
"projects":
    [
     {"project_name":"project1", "completed":false},
     {"project_name":"project2", "completed":true}
    ]
}
```

  Esempio 1

  La seguente query restituisce questi risultati:

```
Select s.name from S3Object s
```

```
{"name":"Susan Smith"}
```

Esempio 2

La seguente query restituisce questi risultati:

```
Select s.projects[0].project_name from S3Object s
```

```
{"project_name":"project1"}
```

Distinzione tra maiuscole e minuscole nei nomi di intestazioni/attributi

Con Amazon S3 Select puoi utilizzare le virgolette doppie per indicare che nelle intestazioni di colonna (per gli oggetti CSV) e negli attributi (per gli oggetti JSON) si applica la distinzione tra maiuscole e minuscole. In assenza delle virgolette doppie, le intestazioni e gli attributi degli oggetti non prevedono distinzione tra maiuscole e minuscole. In caso di ambiguità viene generato un errore.

I seguenti esempi sono 1) oggetti Amazon S3 in formato CSV con le intestazioni di colonna specificate e con impostato su `"Use"` per la richiesta di query oppure 2) oggetti Amazon S3 in formato JSON con gli attributi specificati.

Esempio #1: l'oggetto su cui viene eseguita la query ha intestazione o attributo `NAME`.

- La seguente espressione restituisce correttamente i valori dall'oggetto. Poiché non sono presenti virgolette, la query non fa distinzione tra maiuscole e minuscole.

  ```
  SELECT s.name from S3Object s
  ```

- La seguente espressione genera un errore 400 `MissingHeaderName`. Poiché sono presenti le virgolette, la query fa distinzione tra maiuscole e minuscole.

  ```
  SELECT s."name" from S3Object s
  ```

Esempio 2: l'oggetto Amazon S3 su cui viene eseguita la query ha un'intestazione o attributo con `NAME` e un'altra intestazione o attributo con `name`.

- La seguente espressione genera un errore 400 `AmbiguousFieldName`. Poiché senza virgolette, senza distinzione tra maiuscole e minuscole, ma ci sono due corrispondenze, quindi viene generato un errore.

```
SELECT s.name from S3Object s
```

- La seguente espressione restituisce correttamente i valori dall'oggetto. Poiché sono presenti le virgolette, la query fa distinzione tra maiuscole e minuscole, quindi non vi è alcuna ambiguità.

```
SELECT s."NAME" from S3Object s
```

Utilizzo di parole chiave riservate come termini definiti dall'utente

Amazon S3 Select dispone di un set di parole chiave riservate, necessarie per eseguire le espressioni SQL utilizzate per le query sul contenuto degli oggetti. Tali parole chiave riservate includono nomi di funzioni, tipi di dati, operatori e così via. In alcuni casi, i termini definiti dall'utente, ad esempio le intestazioni di colonna (per i file CSV) o gli attributi (per gli oggetti JSON) possono essere in conflitto con una parola chiave riservata. Quando ciò si verifica, devi utilizzare le virgolette doppie per indicare che stai intenzionalmente utilizzando un termine definito dall'utente in conflitto con una parola chiave riservata. In caso contrario, verrà generato un errore di analisi 400.

Per l'elenco completo delle parole chiave riservate, consulta Parole chiave riservate.

Il seguente esempio è 1) un oggetto Amazon S3 in formato CSV con le intestazioni di colonna specificate e con `FileHeaderInfo` impostato su `"Use"` per la richiesta di query oppure 2) un oggetto Amazon S3 in formato JSON con gli attributi specificati.

Esempio: l'oggetto su cui viene eseguita la query ha un'intestazione o attributo denominato `CAST`, che è una parola chiave riservata.

- La seguente espressione restituisce correttamente i valori dall'oggetto. Poiché nella query vengono utilizzate le virgolette, S3 Select utilizza l'intestazione o l'attributo definiti dall'utente.

```
SELECT s."CAST" from S3Object s
```

- La seguente espressione genera un errore 400 di analisi. Poiché nella query non vengono utilizzate virgolette, `CAST` entra in conflitto con una parola chiave riservata.

```
SELECT s.CAST from S3Object s
```

Espressioni scalari

Nella clausola `WHERE` e nell'elenco `SELECT`, puoi avere espressioni scalari di SQL, ovvero espressioni che restituiscono valori scalari. Queste espressioni hanno la seguente forma:

- ***literal***

  Un valore letterale SQL.

- ***column_reference***

  Un riferimento a una colonna nel modulo *column_name* o *alias*`.`*column_name*.

- ***unary_op expression***

  In questo caso, ***unary_op*** è un operatore SQL unario.

- ***expression binary_op expression***

  In questo caso, ***binary_op*** è un operatore binario SQL.

- ***func_name***

  In questo caso, ***func_name*** è il nome della funzione scalare da richiamare.

- ***expression*** `[ NOT ] BETWEEN` ***expression*** `AND` ***expression***
- ***expression*** `LIKE` ***expression*** `[ ESCAPE` ***expression*** `]`

Tipi di dati

Amazon S3 Select supporta diversi tipi di dati primitivi.

Conversioni dei tipi di dati

La regola generale è di seguire la funzione `CAST` se definita. Se `CAST` non è definita, tutti i dati di input vengono trattati come stringa. In tal caso, è necessario inserire i dati di input ai tipi di dati pertinenti quando necessario.

Per ulteriori informazioni sulla funzione `CAST`, consulta CAST.

Tipi di dati supportati

Amazon S3 Select supporta il seguente set di tipi di dati primitivi.

Nome	Descrizione	Esempi
`bool`	Un valore booleano, `TRUE` o `FALSE`.	`FALSE`
`int`, `integer`	Intero con segno da 8 byte compreso nell'intervallo da -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807.	`100000`
`string`	Stringa di lunghezza variabile con codifica UTF8. Il limite di default è 1 carattere. Il limite massimo di caratteri è 2.147.483.647.	`'xyz'`
`float`	Numero in virgola mobile a 8 byte.	`CAST(0.456 AS FLOAT)`
`decimal`, `numeric`	Numero in base 10, con una precisione massima di 38 (ovvero il numero massimo di cifre significative) e con scala compresa nell'intervallo da -2^{31} a $2^{31}-1$ (ovvero l'esponente in base 10). **Note** Amazon S3 Select ignora il dimensionamento e la precisione quando vengono forniti entrambi contemporaneamente.	`123.456`
`timestamp`	I time stamp rappresentano un momento specifico nel tempo, includono sempre un offset locale e consentono di stabilire una precisione arbitraria. Nel formato di testo, i timestamp seguono i formati di data e ora della notazione W3C, ma devono terminare con il letterale T se la precisione non è di almeno un giorno completo. Le frazioni di secondo sono consentite, con una precisione di almeno una cifra e un valore massimo illimitato. Gli offset in ora locale possono essere rappresen	`CAST('2007-04-05T14:30Z' AS TIMESTAMP)`

Nome	Descrizione	Esempi
	tati con il formato UTC ora:minuto o con il letterale Z per indicare un'ora locale UTC. Le differenze per l'ora locale sono obbligatorie nei time stamp che includono l'ora e non sono consentiti nei valori di data.	

Tipi supportati Parquet

Amazon S3 Select supporta i seguenti tipi di Parquet:

- `DATE`
- `DECIMAL`
- `ENUM`
- `INT(8)`
- `INT(16)`
- `INT(32)`
- `INT(64)`
- `LIST`

> **Note**
>
> Per l'output del tipo `LIST` Parquet, Amazon S3 Select supporta solo il formato JSON. Tuttavia, se la query limita i dati a valori semplici, il tipo di `LIST` Parquet può essere interrogato anche in formato CSV.

- `STRING`
- `TIMESTAMP` precisione supportata (`MILLIS`/`MICROS`/`NANOS`)

> **Note**
>
> I timestamp salvati come `INT(96)` non sono supportati.
> A causa della gamma del tipo `INT(64)`, i timestamp che utilizzano l'unità `NANOS` possono rappresentare solo valori compresi tra `1677-09-21 00:12:43` e `2262-04-11`

23:47:16. I valori al di fuori di questo intervallo non possono essere rappresentati con l'unità NANOS.

Mappatura dei tipi di Parquet ai tipi di dati supportati in Amazon S3 Select

Tipi di Parquet	Tipi di dati supportati
DATE	timestamp
DECIMAL	decimal, numeric
ENUM	string
INT(8)	int, integer
INT(16)	int, integer
INT(32)	int, integer
INT(64)	decimal, numeric
LIST	Ogni tipo di Parquet nell'elenco è mappato al tipo di dati corrispondente
STRING	string
TIMESTAMP	timestamp

Operatori

Amazon S3 Select supporta i seguenti operatori.

Operatori logici

- `AND`
- `NOT`
- `OR`

Operatori di confronto

- `<`
- `>`
- `<=`
- `>=`
- `=`
- `<>`
- `!=`
- `BETWEEN`
- `IN` – Ad esempio: `IN ('a', 'b', 'c')`

Operatori di criteri di ricerca

- `LIKE`
- `_` (Corrisponde a qualsiasi carattere)
- `%` (Corrisponde a qualsiasi sequenza di caratteri)

Operatori unitari

- `IS NULL`
- `IS NOT NULL`

Operatori matematici

Sono supportati gli operatori di addizione, sottrazione, moltiplicazione, divisione e modulo come segue:

- +
- -
- *
- /
- %

Precedenza degli operatori

Nella tabella seguente è indicata la precedenza degli operatori in ordine decrescente.

Operatore o elemento	Associatività	Campo obbligatorio
-	destra	meno unario
*, /, %	sinistra	moltiplicazione, divisione, modulo
+, -	sinistra	addizione, sottrazione
IN		appartenenza a un set
BETWEEN		limitazione a un intervallo
LIKE		criteri di ricerca di stringhe
<>		minore di, maggiore di
=	destra	uguaglianza, assegnazione

Operatore o elemento	Associatività	Campo obbligatorio
NOT	destra	negazione logica
AND	sinistra	congiunzione logica
OR	sinistra	disgiunzione logica

Parole chiave riservate

Di seguito è riportato l'elenco delle parole chiave riservate per Amazon S3 Select. Sono inclusi nomi di funzioni, tipi di dati, operatori e così via, necessari per eseguire le espressioni SQL utilizzate per le query sul contenuto degli oggetti.

```
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
```

Funzioni SQL

Amazon S3 Select supporta le seguenti funzioni SQL.

Argomenti

Funzioni di aggregazione

Amazon S3 Select supporta le seguenti funzioni di aggregazione:

Funzione	Tipo di argomento	Tipo restituito
`AVG(`*expression*`)`	`INT`, `FLOAT`, `DECIMAL`	`DECIMAL` per un argomento `INT`, `FLOAT` un argomento in virgola mobile; in caso contrario, lo stesso tipo di dati dell'argomento.
`COUNT`	-	`INT`
`MAX(`*expression*`)`	`INT`, `DECIMAL`	Lo stesso tipo dell'argomento.
`MIN(`*expression*`)`	`INT`, `DECIMAL`	Lo stesso tipo dell'argomento.
`SUM(`*expression*`)`	`INT`, `FLOAT`, `DOUBLE`, `DECIMAL`	`INT` per un argomento `INT`, `FLOAT`

Funzione	Tipo di argomento	Tipo restituito
		un argomento in virgola mobile; in caso contrario, lo stesso tipo di dati dell'argo mento.

SUM Esempio

Per aggregare le dimensioni totali degli oggetti di una cartella in un report S3 Inventory, usa un'espressione SUM.

Il seguente report S3 Inventory è un file CSV compresso con GZIP. Sono disponibili tre colonne.

- La prima colonna è il nome del bucket S3 (*DOC-EXAMPLE-BUCKET*) a cui è destinato il rapporto S3 Inventory.
- La seconda colonna è il nome della chiave dell'oggetto che identifica in modo univoco l'oggetto nel bucket.

  Il valore *example-folder/* nella prima riga si riferisce alla cartella *example-folder*. Quando crei una cartella nel bucket in Amazon S3, S3 crea un oggetto con dimensioni pari a 0 byte con una chiave impostata sul nome della cartella fornito.

  Il valore *example-folder/object1* nella seconda riga si riferisce all'oggetto *object1* nella cartella *example-folder*.

  Il valore *example-folder/object2* nella terza riga si riferisce all'oggetto *object2* nella cartella *example-folder*.

  Per ulteriori informazioni sulle cartelle S3, consulta Organizzazione degli oggetti nella console di Amazon S3 utilizzando le cartelle.
- La terza colonna è la dimensione dell'oggetto in byte.

```
"DOC-EXAMPLE-BUCKET","example-folder/","0"
"DOC-EXAMPLE-BUCKET","example-folder/object1","2011267"
```

```
"DOC-EXAMPLE-BUCKET","example-folder/object2","1570024"
```

Per utilizzare un'espressione SUM per calcolare la dimensione totale della cartella *example-folder*, esegui la query SQL con Amazon S3 Select.

```
SELECT SUM(CAST(_3 as INT)) FROM s3object s WHERE _2 LIKE 'example-folder/%' AND _2 !=
 'example-folder/';
```

Risultato della query:

```
3581291
```

Funzioni condizionali

Amazon S3 Select supporta le seguenti funzioni condizionali.

Argomenti

- CASE
- COALESCE
- NULLIF

CASE

L'espressione CASE è un'espressione condizionale, simile alle istruzioni `if/then/else` presenti in altre lingue. CASE è utilizzata per specificare un risultato quando ci sono condizioni multiple. Esistono due tipi di espressioni CASE: semplici e ricercate.

Nelle espressioni CASE semplici, un'espressione viene confrontata con un valore. Quando viene trovata una corrispondenza, viene applicata l'azione specificata nella clausola THEN. Se non viene trovata una corrispondenza, viene applicata l'azione nella clausola ELSE.

Nelle espressioni CASE cercate, ogni CASE viene valutata in base a un'espressione booleana e l'istruzione CASE restituisce la prima CASE corrispondente. Se non vengono trovate corrispondenze CASE tra le clausole WHEN, viene restituita l'operazione nella clausola ELSE.

Sintassi

> **Note**
>
> Attualmente Amazon S3 Select non supporta `ORDER BY` o query che contengono nuove righe. Assicurati di utilizzare query senza interruzioni di riga.

Quella che segue è una semplice dichiarazione `CASE` che viene utilizzata per soddisfare le condizioni:

```
CASE expression WHEN value THEN result [WHEN...] [ELSE result] END
```

Di seguito è disponibile una dichiarazione `CASE` ricercata che viene utilizzata per valutare ogni condizione:

```
CASE WHEN boolean condition THEN result [WHEN ...] [ELSE result] END
```

Esempi

> **Note**
>
> Se utilizzi la console Amazon S3 per eseguire i seguenti esempi e il file CSV contiene una riga di intestazione, seleziona Exclude the first line of CSV data (Escludi la prima riga di dati CSV).

Esempio 1: utilizza una semplice espressione `CASE` per sostituire `New York City` con `Big Apple` in una query. Sostituire tutti gli altri nomi di città con `other`.

```
SELECT venuecity, CASE venuecity WHEN 'New York City' THEN 'Big Apple' ELSE 'other' END
 FROM S3Object;
```

Risultato della query:

```
venuecity        |   case
-----------------+-----------
Los Angeles      | other
New York City    | Big Apple
```

```
San Francisco    | other
Baltimore        | other
...
```

Esempio 2: utilizza un'espressione CASE con ricerca per assegnare numeri di gruppo in base al valore `pricepaid` per le vendite di biglietti singoli:

```
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;
```

Risultato della query:

```
pricepaid |  case
-----------+---------
12624.00 | group 2
10000.00 | group 3
10000.00 | group 3
9996.00 | group 1
9988.00 | group 1
...
```

COALESCE

COALESCE valuta gli argomenti in ordine e restituisce il primo valore non sconosciuto, ovvero il primo non nullo o non mancante. Questa funzione non propaga valori null e mancanti.

Sintassi

```
COALESCE ( expression, expression, ... )
```

Parametri

expression

L'espressione di destinazione su cui viene eseguita la funzione.

Esempi

```
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

Date due espressioni, `NULLIF` restituisce `NULL` se le due espressioni restituiscono lo stesso valore. In caso contrario, restituisce il risultato della valutazione della prima espressione.

Sintassi

```
NULLIF ( expression1, expression2 )
```

Parametri

expression1, *expression2*

Le espressioni di destinazione su cui viene eseguita la funzione.

Esempi

```
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
```

Funzioni di conversione

Amazon S3 Select supporta le seguenti funzioni di conversione.

Argomenti

- CAST

CAST

La funzione CAST converte un'entità, ad esempio un'espressione che restituisce un singolo valore, da un tipo a un altro.

Sintassi

```
CAST ( expression AS data_type )
```

Parametri

expression

Una combinazione di uno o più valori, operatori e funzioni SQL che restituisce un valore.

data_type

Il tipo di dati di destinazione, ad esempio INT, per il quale eseguire il cast dell'espressione. Per un elenco dei tipi di dati supportati, consulta Tipi di dati.

Esempi

```
CAST('2007-04-05T14:30Z' AS TIMESTAMP)
CAST(0.456 AS FLOAT)
```

Funzioni di data

Amazon S3 Select supporta le seguenti funzioni di data.

Argomenti

- DATE_ADD
- DATE_DIFF
- EXTRACT
- TO_STRING
- TO_TIMESTAMP
- UTCNOW

DATE_ADD

Dati una parte di data, una quantità e un timestamp, restituisce un timestamp aggiornato modificando la parte di data in base alla quantità.

Sintassi

```
DATE_ADD( date_part, quantity, timestamp )
```

Parametri

date_part

Specifica la parte di data da modificare. Può essere una delle seguenti:

- anno
- mese
- giorno
- ora
- minuti
- secondo

quantity

Il valore da applicare al timestamp aggiornato. I valori positivi per *quantity* vengono aggiunti a date_part del timestamp, mentre i valori negativi vengono sottratti.

timestamp

Il timestamp di destinazione su cui viene eseguita la funzione.

Esempi

```
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

Dati una parte di data e due timestamp validi, `DATE_DIFF` restituisce la differenza in parti di data. Il valore restituito è un numero intero negativo quando il valore *`date_part`* di *`timestamp1`* è maggiore del valore *`date_part`* di *`timestamp2`*. Il valore restituito è un numero intero positivo quando il valore *`date_part`* di *`timestamp1`* è minore del valore *`date_part`* di *`timestamp2`*.

Sintassi

```
DATE_DIFF( date_part, timestamp1, timestamp2 )
```

Parametri

`date_part`

Specifica la parte dei timestamp da confrontare. Per la definizione di `date_part`, consulta DATE_ADD.

`timestamp1`

Il primo timestamp da confrontare.

`timestamp2`

Il secondo timestamp da confrontare.

Esempi

```
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

Dati una parte di data e un timestamp, `EXTRACT` restituisce il valore della parte di data del timestamp.

Sintassi

```
EXTRACT( date_part FROM timestamp )
```

Parametri

date_part

Specifica la parte dei timestamp da estrarre. Può essere una delle seguenti:

- YEAR
- MONTH
- DAY
- HOUR
- MINUTE
- SECOND
- TIMEZONE_HOUR
- TIMEZONE_MINUTE

timestamp

Il timestamp di destinazione su cui viene eseguita la funzione.

Esempi

```
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

Dati un timestamp e un pattern di formato, TO_STRING restituisce una rappresentazione in formato stringa del timestamp fornito nel formato specificato.

Sintassi

```
TO_STRING ( timestamp time_format_pattern )
```

Parametri

timestamp

Il timestamp di destinazione su cui viene eseguita la funzione.

time_format_pattern

Una stringa che ha le seguenti interpretazioni di caratteri speciali.

Formato	Esempio	Descrizione
yy	69	Anno a 2 cifre
y	1969	Anno a 4 cifre
yyyy	1969	Anno a 4 cifre con l'aggiunta di zero
M	1	Mese dell'anno
MM	01	Mese dell'anno con l'aggiunta di zero
MMM	Jan	Nome abbreviato del mese dell'anno
MMMM	January	Nome completo

Formato	Esempio	Descrizione
		del mese dell'anno
MMMMM	J	Prima lettera del mese dell'anno (NOTA: questo formato non è utilizzabile con la funzione TO_TIMESTAMP .)
d	2	Giorno del mese (1-31)
dd	02	Giorno del mese con l'aggiunta di zero (01-31)
a	AM	AM o PM
h	3	Ora del giorno (1-12)
hh	03	Ora del giorno con l'aggiunta di zero (01-12)
H	3	Ora del giorno (0-23)
HH	03	Ora del giorno con l'aggiunta di zero (00-23)

Formato	Esempio	Descrizione
m	4	Minuto dell'ora (0-59)
mm	04	Minuto dell'ora con l'aggiunta di zero (00-59)
s	5	Secondo del minuto (0-59)
ss	05	Secondo del minuto con l'aggiunta di zero (00-59)
S	0	Frazione di secondo (precisione: 0,1, intervallo: 0,0-0,9)
SS	6	Frazione di secondo (precisione: 0,01, intervallo: 0,0-0,99)
SSS	60	Frazione di secondo (precisione: 0,001, intervallo: 0,0-0,999)
…	…	…

Formato	Esempio	Descrizione
SSSSSSSSS	60000000	Frazione di secondo (precisione massima: 1 nanosecondo, intervallo: 0,0-0,999999999)
n	60000000	Nano di secondo
X	+07 o Z	Offset in ore o Z se l'offset è 0
XX o XXXX	+0700 o Z	Offset in ore e minuti o Z se l'offset è 0
XXX o XXXXX	+07:00 o Z	Offset in ore e minuti o Z se l'offset è 0
x	7	Offset in ore
xx o xxxx	700	Offset in ore e minuti
xxx o xxxxx	+07:00	Offset in ore e minuti

Esempi

```
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

Data una stringa, `TO_TIMESTAMP` la converte in un timestamp. `TO_TIMESTAMP` è l'operazione inversa di `TO_STRING`.

Sintassi

```
TO_TIMESTAMP ( string )
```

Parametri

`string`

La stringa di destinazione su cui viene eseguita la funzione.

Esempi

```
TO_TIMESTAMP('2007T')                          -- `2007T`
TO_TIMESTAMP('2007-02-23T12:14:33.079-08:00') -- `2007-02-23T12:14:33.079-08:00`
```

UTCNOW

Restituisce l'ora corrente in UTC come timestamp.

Sintassi

```
UTCNOW()
```

Parametri

`UTCNOW` non prende parametri.

Esempi

```
UTCNOW() -- 2017-10-13T16:02:11.123Z
```

Funzioni stringa

Amazon S3 Select supporta le seguenti funzioni di stringa.

Argomenti

- CHAR_LENGTH, CHARACTER_LENGTH
- LOWER
- SUBSTRING
- TRIM
- UPPER

CHAR_LENGTH, CHARACTER_LENGTH

`CHAR_LENGTH` (o `CHARACTER_LENGTH`) conta il numero di caratteri della stringa specificata.

> **Note**
>
> `CHAR_LENGTH` e `CHARACTER_LENGTH` sono sinonimi.

Sintassi

```
CHAR_LENGTH ( string )
```

Parametri

`string`

La stringa di destinazione su cui viene eseguita la funzione.

Esempi

```
CHAR_LENGTH('')          -- 0
CHAR_LENGTH('abcdefg')   -- 7
```

LOWER

Data una stringa, `LOWER` converte tutti i caratteri maiuscoli in minuscoli. I caratteri non maiuscoli rimangono invariati.

Sintassi

```
LOWER ( string )
```

Parametri

`string`

La stringa di destinazione su cui viene eseguita la funzione.

Esempi

```
LOWER('AbCdEfG!@#$') -- 'abcdefg!@#$'
```

SUBSTRING

Dati una stringa, un indice iniziale e, facoltativamente, una lunghezza, `SUBSTRING` restituisce la sottostringa dall'indice iniziale fino alla fine della stringa oppure fino alla lunghezza specificata.

> **Note**
>
> Il primo carattere della stringa di input ha indice 1.
>
> - Se `start` è < 1, senza una lunghezza specificata allora viene impostato su 1.
> - Se `start` è < 1, con una lunghezza specificata, allora la posizione dell'indice viene impostata su `start + length -1`.
> - Se `start + length -1` < 0 allora viene restituita una stringa vuota.
> - Se `start + length -1` > = 0 allora viene restituita la sottostringa che inizia dall'indice 1 con lunghezza `start + length - 1`.

Sintassi

```
SUBSTRING( string FROM start [ FOR length ] )
```

Parametri

string

La stringa di destinazione su cui viene eseguita la funzione.

start

La posizione iniziale della stringa.

length

La lunghezza della sottostringa da restituire. Se non è presente, procede fino alla fine della stringa.

Esempi

```
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

Taglia i caratteri iniziali o finali di una stringa. Il carattere di default da rimuovere è uno spazio (' ').

Sintassi

```
TRIM ( [[LEADING | TRAILING | BOTH remove_chars] FROM] string )
```

Parametri

string

La stringa di destinazione su cui viene eseguita la funzione.

`LEADING | TRAILING | BOTH`

Il parametro indica se tagliare i caratteri iniziali o finali o entrambi.

`remove_chars`

Il set di caratteri da rimuovere. *`remove_chars`* può essere una stringa con lunghezza > 1. Questa funzione restituisce la stringa da cui sono stati rimossi i caratteri specificati in *`remove_chars`* trovati all'inizio o alla fine della stringa.

Esempi

```
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

Data una stringa, UPPER converte tutti i caratteri minuscoli in maiuscoli. I caratteri non minuscoli rimangono invariati.

Sintassi

```
UPPER ( string )
```

Parametri

`string`

La stringa di destinazione su cui viene eseguita la funzione.

Esempi

```
UPPER('AbCdEfG!@#$') -- 'ABCDEFG!@#$'
```

Esecuzione di operazioni in batch su larga scala su oggetti Amazon S3

Puoi utilizzare le operazioni in batch S3 per eseguire operazioni in batch su vasta scala su oggetti Amazon S3. Le operazioni in batch S3 possono eseguire una singola operazione su elenchi di oggetti Amazon S3 specificati. Un solo processo può eseguire l'operazione specificata su miliardi di oggetti contenenti exabyte di dati. Amazon S3 tiene traccia dell'avanzamento, invia notifiche e conserva un report di completamento dettagliato di tutte le operazioni, offrendo un'esperienza serverless verificabile e completamente gestita. Puoi utilizzare le operazioni in batch S3 tramite la AWS Management Console, AWS CLI, gli SDK Amazon o l'API REST.

Le operazioni in batch S3 permettono di copiare oggetti e impostare i tag dell'oggetto o le liste di controllo accessi (ACL). Puoi anche avviare ripristini di oggetti da Amazon S3 Glacier Flexible Retrieval o richiamare una funzione AWS Lambda per eseguire operazioni personalizzate utilizzando i tuoi oggetti. Puoi eseguire queste operazioni su un elenco personalizzato di oggetti oppure utilizzare un report di Amazon S3 Inventory per generare facilmente liste di oggetti. Poiché le operazioni in batch Amazon S3 utilizzano le stesse API di Amazon S3 che già utilizzi con Amazon S3, l'interfaccia ti sarà familiare.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory. Per ulteriori informazioni sull'utilizzo delle Operazioni in batch con S3 Express One Zone e bucket di directory, consulta Utilizzo di Operazioni in batch con S3 Express One Zone.

Nozioni di base sulle operazioni in batch S3

Puoi utilizzare le operazioni in batch S3 per eseguire operazioni in batch su vasta scala su oggetti Amazon S3. Le operazioni in batch S3 possono eseguire una singola operazione su elenchi di oggetti Amazon S3 specificati.

Terminologia

In questa sezione vengono utilizzati i termini processo, operazione e attività, definiti di seguito:

Processo

Un processo è l'unità di lavoro di base per le operazioni in batch S3. Un processo include tutte le informazioni necessarie per eseguire l'operazione specificata sugli oggetti elencati nel file manifest Una volta fornite queste informazioni e richiesto l'inizio del processo, il processo esegue l'operazione specificata su ciascun oggetto del manifest.

Operazione

L'operazione è il tipo di operazione API, ad esempio la copia di oggetti, che desideri venga eseguita dal processo Batch Operations. Ogni processo esegue un singolo tipo di operazione in tutti gli oggetti specificati nel manifest.

Attività

Un'attività è l'unità di esecuzione per un processo. Un'attività rappresenta una singola chiamata a un'operazione API Amazon S3 o AWS Lambda per eseguire l'operazione del processo su un singolo oggetto. Nel corso del ciclo di vita di un processo, le operazioni in batch S3 creano un'unica attività per ogni oggetto specificato nel manifest.

Funzionamento di un processo Batch S3 Operations

Un processo è l'unità di lavoro di base per le operazioni in batch S3. Un processo include tutte le informazioni necessarie per eseguire l'operazione specificata su un elenco di oggetti. Per creare un processo, devi fornire alle operazioni in batch S3 un elenco di oggetti e specificare l'operazione da eseguire su tali oggetti.

Per informazioni sulle operazioni in batch supportate da S3, consulta Operazioni supportate dalle operazioni in batch S3.

Un processo in batch esegue l'operazione specificata su ciascun oggetto incluso nel manifest. Un manifest elenca gli oggetti che si desidera elaborare con un processo batch e viene memorizzato come oggetto in un bucket. Puoi utilizzare report in formato CSV (comma-separated values, valori separati da virgola) Amazon S3 Inventory come manifest per semplificare la creazione di elenchi di oggetti di grandi dimensioni presenti in un bucket. È anche possibile specificare un manifest in un formato CSV semplice che consente di eseguire operazioni batch su un elenco personalizzato di oggetti contenuti in un singolo bucket.

Dopo aver creato un processo, Amazon S3 elabora l'elenco di oggetti nel manifest ed esegue l'operazione specificata su ogni oggetto. Durante l'esecuzione di un processo, puoi monitorarne lo

stato a livello di programmazione o tramite la console Amazon S3. È anche possibile configurare un processo affinché generi un rapporto di completamento al termine della sua esecuzione. Il rapporto di completamento descrive i risultati di ciascuna attività eseguita dal processo. Per ulteriori informazioni sul monitoraggio dei processi, consulta Gestione dei processi di operazioni in batch Amazon S3.

Tutorial sulle operazioni in batch S3

Il seguente tutorial presenta end-to-end le procedure complete per alcune attività di Batch Operations.

- Tutorial: transcodifica in batch di video con S3 Batch Operations e AWS LambdaAWS Elemental MediaConvert

Concessione delle autorizzazioni per le operazioni in batch Amazon S3

Prima di creare ed eseguire processi operazioni in batch S3, è necessario concedere le autorizzazioni necessarie. Per creare un processo di operazioni in batch Amazon S3, è necessaria l'autorizzazione utente `s3:CreateJob`. La stessa entità che crea il job deve inoltre avere l'`iam:PassRole`autorizzazione a passare il ruolo AWS Identity and Access Management (IAM) specificato per il job a Batch Operations.

Per informazioni generali sulla specifica delle risorse IAM, consulta Elementi delle policy IAM JSON e Resource nella Guida per l'utente IAM. Nelle sezioni seguenti vengono fornite informazioni sulla creazione di un ruolo IAM e sul collegamento delle policy.

Argomenti

- Creazione di un ruolo IAM di operazioni in batch S3
- Allegare policy di autorizzazione

Creazione di un ruolo IAM di operazioni in batch S3

Perché Amazon S3 possa eseguire operazioni in batch S3 per tuo conto, occorre concedergli le opportune autorizzazioni. Queste autorizzazioni vengono concesse tramite un ruolo AWS Identity and Access Management (IAM). Questa sezione fornisce esempi delle policy di attendibilità e di autorizzazione che si possono usare quando si crea un ruolo IAM. Per ulteriori informazioni, consulta Ruoli IAM nella Guida per l'utente IAM. Per alcuni esempi, consulta Controllo delle autorizzazioni per S3 Batch Operations utilizzando i tag di processo e Copia di oggetti mediante operazioni in batch S3.

Nelle policy IAM è inoltre possibile utilizzare le chiavi di condizione per filtrare le autorizzazioni di accesso per i processi di operazioni in batch Amazon S3. Per ulteriori informazioni e un elenco completo delle chiavi di condizione specifiche di Amazon S3, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.

Policy di trust

Per consentire al principale del servizio di operazioni in batch S3 di assumere il ruolo IAM, collega la seguente policy di attendibilità al ruolo.

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

Allegare policy di autorizzazione

A seconda del tipo di operazioni, puoi collegare una delle policy seguenti.

Prima di configurare le autorizzazioni, tieni presente quanto segue:

- A prescindere dall'operazione, Amazon S3 necessita delle autorizzazioni per leggere l'oggetto manifest dal bucket S3 e, facoltativamente, per scrivere un report nel bucket. Quindi, tutte le policy seguenti includono queste autorizzazioni.
- Per i manifest di report di Amazon S3 Inventory, S3 Batch Operations richiede l'autorizzazione per leggere l'oggetto manifest.json e tutti i file di dati CSV associati.
- Autorizzazioni specifiche della versione come `s3:GetObjectVersion` sono richieste solo quando si specifica l'ID versione degli oggetti.
- Se esegui S3 Batch Operations su oggetti crittografati, il ruolo IAM deve avere accesso anche alle AWS KMS chiavi utilizzate per crittografarli.

- Se invii un manifesto del report di inventario crittografato con AWS KMS, la tua policy IAM deve includere le autorizzazioni `"kms:GenerateDataKey"` per l'oggetto manifest.json `"kms:Decrypt"` e tutti i file di dati CSV associati.
- Se il processo Batch Operations genera un manifest in un bucket con ACL abilitati e si trova in un AWS account diverso, è necessario concedere l'`s3:PutObjectAcl`autorizzazione nella policy IAM del ruolo IAM configurato per il processo batch. Se non si include questa autorizzazione, il processo batch ha esito negativo e viene visualizzato l'errore. `Error occurred when preparing manifest: Failed to write manifest`

Copia oggetti: 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/*"
            ]
        }
    ]
}
```

Sostituisci l'etichettatura degli oggetti: 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/*"
      ]
    }
  ]
}
```

Elimina l'etichettatura degli oggetti: 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/*"
            ]
        }
```

```
    ]
}
```

Sostituisci l'elenco di controllo degli accessi: 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/*"
      ]
    }
  ]
}
```

Ripristina oggetti: 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/*"
      ]
    }
  ]
}
```

Applica la conservazione di Object Lock: PutObjectRetention

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetBucketObjectLockConfiguration",
            "Resource": [
                "arn:aws:s3:::TargetResource"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "s3:PutObjectRetention",
                "s3:BypassGovernanceRetention"
            ],
            "Resource": [
                "arn:aws:s3:::TargetResource/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::ManifestBucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::ReportBucket/*"
            ]
        }
    ]
}
```

Applica la conservazione legale di Object Lock: PutObjectLegalHold

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetBucketObjectLockConfiguration",
            "Resource": [
                "arn:aws:s3:::TargetResource"
            ]
        },
        {
```

```
            "Effect": "Allow",
            "Action": "s3:PutObjectLegalHold",
            "Resource": [
                "arn:aws:s3:::TargetResource/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::ManifestBucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::ReportBucket/*"
            ]
        }
    ]
}
```

Replica oggetti esistenti: InitiateReplication con un manifesto generato da S3

Utilizza questa policy se utilizzi e archivia un manifesto generato da S3. Per ulteriori informazioni sull'utilizzo di Batch Operations per la replica di oggetti esistenti, consulta la sezione Replica di oggetti esistenti con S3 Batch Replication.

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

Replica oggetti esistenti: InitiateReplication con un manifesto utente

Utilizza questa policy se impieghi un manifesto fornito dall'utente. Per ulteriori informazioni sull'utilizzo di Batch Operations per la replica di oggetti esistenti, consulta la sezione Replica di oggetti esistenti con S3 Batch Replication.

```
{
```

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

Creazione di un processo di operazioni in batch S3

Con Operazioni in batch Amazon S3, puoi eseguire operazioni in batch su larga scala su un elenco di oggetti Amazon S3 specifici. In questa sezione vengono descritte le informazioni necessarie per creare un processo S3 Batch Operations e i risultati di una richiesta `CreateJob`. Fornisce inoltre istruzioni per creare un processo Batch Operations utilizzando la console Amazon S3, AWS Command Line Interface (AWS CLI) e. AWS SDK for Java

Quando crei un processo S3 Batch Operations, puoi richiedere un report di completamento per tutte le attività o solo per le attività fallite. Se almeno un'attività è stata richiamata correttamente, Operazioni in batch S3 genera un report per i processi che sono stati completati, che non sono andati a buon fine o che sono stati annullati. Per ulteriori informazioni, consulta Esempi: report di completamento delle operazioni in batch S3.

Argomenti

Elementi della richiesta di un processo di operazioni in batch

Per creare un processo di operazioni in batch S3, è necessario fornire le seguenti informazioni:

Operazione

Specifica l'operazione che vuoi far eseguire alle operazioni in batch S3 sugli oggetti nel manifest. Ogni tipo di operazione accetta parametri specifici di tale operazione. Con Batch Operations, è possibile eseguire un'operazione in blocco, con gli stessi risultati che si otterrebbe se si eseguisse tale operazione one-by-one su ciascun oggetto.

Manifest

Il manifesto è un elenco di tutti gli oggetti sui quali Operazioni in batch S3 esegue l'operazione desiderata. Per specificare un manifesto per un processo Operazioni in batch, puoi utilizzare i seguenti metodi:

- Crea manualmente l'elenco di oggetti personalizzato in formato CSV.
- Scegli un report Amazon S3 Inventory esistente in formato CSV.
- Indirizza Operazioni in batch per generare automaticamente un manifesto in base ai criteri di filtro degli oggetti specificati quando si crea il processo. Questa opzione è disponibile per i lavori di replica in batch creati nella console Amazon S3 o per qualsiasi tipo di lavoro creato utilizzando gli SDK o AWS CLI l'API AWS REST di Amazon S3.

> **Note**
>
> - A prescindere dalla modalità di specifica del manifesto, l'elenco stesso deve essere archiviato in un bucket per uso generico. Operazioni in batch non è in grado di importare i manifesti esistenti da, o salvare i manifesti generati in, bucket di directory.

Gli oggetti descritti all'interno del manifesto, tuttavia, possono essere archiviati in bucket di directory. Per ulteriori informazioni, consulta Directory buckets.

- Se gli oggetti nel manifesto si trovano in un bucket con versione, la specifica degli ID versione per gli oggetti indirizza Operazioni in batch a eseguire l'operazione su una versione specifica. Se nessun ID versione è specificato, Operazioni in batch esegue l'operazione sulla versione più recente degli oggetti. Se il manifesto include un campo ID versione, è necessario fornire un ID versione per tutti gli oggetti del manifesto.

Per ulteriori informazioni, consulta Specifica di un manifest.

Priorità

Utilizza le priorità del processo per indicarne la priorità rispetto agli altri processi in esecuzione sul tuo account. Numeri maggiori indicano una priorità più alta.

Le priorità del lavoro hanno un significato solo rispetto alle priorità stabilite per altri lavori nello stesso account e nella stessa regione. Pertanto puoi scegliere qualsiasi sistema di numerazione utile. Ad esempio, potrebbe essere necessario assegnare a tutti i processi Ripristina (`RestoreObject`) una priorità di 1, a tutti i processi Copia (`CopyObject`) una priorità di 2 e a tutti i processi Sostituisci liste di controllo degli accessi (ACL) (`PutObjectAcl`) una priorità di 3.

Operazioni in batch S3 assegna la priorità ai processi in base ai numeri di priorità ma non è garantito un ordinamento rigoroso. Pertanto, si consiglia di non utilizzare le priorità dei processi per accertarsi che un processo inizi o termini prima di un altro. Per essere certo che l'ordine venga rigidamente rispettato, attendi che un processo sia terminato prima di iniziare quello successivo.

RoleArn

Specificate un ruolo AWS Identity and Access Management (IAM) per eseguire il job. Il ruolo IAM utilizzato deve avere le autorizzazioni necessarie per eseguire l'operazione specificata nel processo. Ad esempio, per eseguire un processo `CopyObject`, il ruolo IAM deve disporre dell'autorizzazione `s3:GetObject` per il bucket di origine e dell'autorizzazione `s3:PutObject` per il bucket di destinazione. Il ruolo ha anche bisogno delle autorizzazioni per leggere il manifest e compilare il report di completamento del processo.

Per ulteriori informazioni sui ruoli IAM, consultare Ruoli IAM nella Guida per l'utente di IAM.

Per ulteriori informazioni sulle autorizzazioni di Amazon S3, consulta la sezione Azioni politiche per Amazon S3.

> **Note**
>
> I processi Operazioni in batch che eseguono azioni su bucket di directory richiedono autorizzazioni specifiche. Per ulteriori informazioni, consulta AWS Identity and Access Management (IAM) for S3 Express One Zone.

Report

Specifica se desideri che le operazioni in batch S3 generino un report di completamento. Se richiedi un report di completamento del lavoro, devi inserire i parametri per il report in questo elemento. Le informazioni necessarie includono:

- Il bucket in cui desideri archiviare il report

  > **Note**
  >
  > Il report deve essere archiviato in un bucket per uso generico. Operazioni in batch non può salvare report in bucket di directory. Per ulteriori informazioni, consulta Directory buckets.

- Il formato del report
- Se desideri che il report includa i dettagli di tutte le attività o solo di quelle fallite
- Una stringa di prefisso (facoltativa)

> **Note**
>
> I report di completamento sono sempre crittografati con chiavi gestite di Amazon S3 (SSE-S3).

Tag (opzionale)

È possibile etichettare e controllare l'accesso ai processi di operazioni in batch Amazon S3 aggiungendo tag. Puoi utilizzare tag per identificare il responsabile del processo Operazioni in batch o controllare in che modo gli utenti interagiscono con processi Operazioni in batch. La presenza dei tag dei lavori può consentire o limitare la capacità di un utente di cancellare un lavoro, attivare un lavoro in stato di conferma o cambiare il livello di priorità di un lavoro. Ad

esempio, puoi concedere a un utente l'autorizzazione per richiamare l'operazione `CreateJob`, purché il processo venga creato con il tag `"Department=Finance"`.

È possibile creare lavori con tag ad essi associati e aggiungere tag ai lavori dopo averli creati.

Per ulteriori informazioni, consulta the section called “Utilizzo dei tag”.

Descrizione (facoltativa)

Per tenere traccia e monitorare il processo, è anche possibile fornire una descrizione di un massimo di 256 caratteri. Amazon S3 include questa descrizione ogni volta che restituisce informazioni su un processo o visualizza i dettagli del processo nella console di Amazon S3. Puoi quindi ordinare e filtrare i processi con facilità in base alle descrizioni che hai assegnato loro. Le descrizioni non devono necessariamente essere univoche, quindi puoi utilizzarle come categorie (ad esempio, "Registro settimanale dei processi Copy") per aiutarti a tenere traccia dei gruppi di processi simili.

Specifica di un manifest

Un manifesto è un oggetto Amazon S3 contenente le chiavi degli oggetti su cui Amazon S3 deve agire. Per fornire un manifesto, puoi utilizzare uno dei seguenti metodi:

- Crea un nuovo file manifesto manualmente.
- Utilizza un manifesto esistente.
- Indirizza Operazioni in batch per generare automaticamente un manifesto in base ai criteri di filtro degli oggetti specificati quando si crea il processo. Questa opzione è disponibile per i lavori di replica in batch creati nella console Amazon S3 o per qualsiasi tipo di lavoro creato utilizzando gli SDK o AWS CLI l'API AWS REST di Amazon S3.

> **Note**
>
> A prescindere dalla modalità di specifica del manifesto, l'elenco stesso deve essere archiviato in un bucket per uso generico. Operazioni in batch non è in grado di importare i manifesti esistenti da, o salvare i manifesti generati in, bucket di directory. Gli oggetti descritti all'interno del manifesto, tuttavia, possono essere archiviati in bucket di directory. Per ulteriori informazioni, consulta Directory buckets.

Creazione di un file manifesto

Per creare un file manifesto manualmente, occorre specificare la chiave dell'oggetto manifesto, l'ETag (tag di entità) e l'ID versione facoltativo in un elenco in formato CSV. I contenuti del manifesto devono essere codificati in formato URL.

Per impostazione predefinita, Amazon S3 utilizza automaticamente la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) per crittografare un manifesto caricato in un bucket Amazon S3. I manifesti che utilizzano la crittografia lato server con chiavi fornite dal cliente (SSE-C) non sono supportati. I manifesti che utilizzano la crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS) sono supportati solo quando utilizzi report di inventario in formato CSV. L'utilizzo di un manifesto creato manualmente con non è supportato. AWS KMS

Il manifest deve contenere il nome del bucket, la chiave dell'oggetto e, facoltativamente, la versione dell'oggetto per ciascun oggetto. Gli altri campi nel manifest non vengono utilizzati dalle operazioni in batch S3.

> **Note**
>
> Se gli oggetti nel manifesto si trovano in un bucket con versione, la specifica degli ID versione per gli oggetti indirizza Operazioni in batch a eseguire l'operazione su una versione specifica. Se nessun ID versione è specificato, Operazioni in batch esegue l'operazione sulla versione più recente degli oggetti. Se il manifesto include un campo ID versione, è necessario fornire un ID versione per tutti gli oggetti del manifesto.

Di seguito è riportato un manifest di esempio in formato CSV senza ID versione.

```
Examplebucket,objectkey1
Examplebucket,objectkey2
Examplebucket,objectkey3
Examplebucket,photos/jpgs/objectkey4
Examplebucket,photos/jpgs/newjersey/objectkey5
Examplebucket,object%20key%20with%20spaces
```

Di seguito è riportato un manifesto di esempio in formato CSV che include ID versione.

```
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
```

Specifica di un file manifesto esistente

Puoi specificare un file manifesto per una richiesta di creazione processo utilizzando uno dei due formati elencati di seguito:

- Report di Inventario Amazon S3: deve essere un report di Inventario Amazon S3 in formato CSV. Devi specificare il file `manifest.json` associato al report di inventario. Per ulteriori informazioni sui report di inventario, consulta Amazon S3 Inventory. Se il report di inventario include gli ID versione, S3 Batch Operations agisce sulle versioni degli oggetti specifiche.

  > **Note**
  >
  > - Operazioni in batch S3 supporta report di inventario CSV con crittografia SSE-KMS.
  > - Se si invia un manifesto del report di inventario con crittografia SSE-KMS, la policy IAM deve includere le autorizzazioni `"kms:Decrypt"` e `"kms:GenerateDataKey"` per l'oggetto `manifest.json` e tutti i file di dati CSV associati.

- File CSV: ogni riga nel file deve includere il nome del bucket, la chiave dell'oggetto e, facoltativamente, la versione dell'oggetto. Le chiavi degli oggetti devono essere codificate in formato URL, come mostrato nei seguenti esempi. Il manifest deve includere gli ID versione di tutti gli oggetti oppure ometterli per tutti gli oggetti. Per ulteriori informazioni sul formato CSV del manifesto, consulta JobManifestSpec nella Documentazione di riferimento delle API Amazon Simple Storage Service.

  > **Note**
  >
  > Operazioni in batch S3 non supporta file manifesto in formato CSV con crittografia SSE-KMS.

> **Important**
>
> Quando si utilizza un manifesto creato manualmente e un bucket con versione, si consiglia di specificare gli ID versione per gli oggetti. Quando crei un processo, S3 Batch Operations

analizza l'intero manifest prima di eseguire il processo. Tuttavia, non esegue una "snapshot" dello stato del bucket.
Poiché i manifesti possono contenere miliardi di oggetti, l'esecuzione dei processi potrebbe richiedere molto tempo, influenzando la versione di un oggetto su cui agisce il processo. Supponi di sovrascrivere un oggetto con una nuova versione durante l'esecuzione di un processo e di non aver specificato un ID versione per tale oggetto. In questo caso, Amazon S3 esegue l'operazione sulla versione più recente dell'oggetto, non sulla versione che esisteva al momento della creazione del processo. L'unico modo per evitare questo comportamento è specificare gli ID versione per gli oggetti elencati nel manifest.

Generazione automatica di un manifesto

Puoi indirizzare Amazon S3 a generare un manifesto automaticamente in base ai criteri di filtro degli oggetti specificati al momento della creazione del processo. Questa opzione è disponibile per i lavori di replica in batch creati nella console Amazon S3 o per qualsiasi tipo di lavoro creato utilizzando gli SDK o AWS CLI l'API AWS REST di Amazon S3. Per ulteriori informazioni su Batch Replication, consulta la sezione Replica di oggetti esistenti con S3 Batch Replication.

Per generare un manifesto automaticamente, specifica i seguenti elementi come parte della richiesta di creazione del processo:

- Informazioni sul bucket contenente gli oggetti di origine, inclusi il proprietario del bucket e il nome della risorsa Amazon (ARN)
- Informazioni sull'output del manifesto, incluso un flag per creare un file manifesto, il proprietario del bucket di output, l'ARN, il prefisso, il formato del file e il tipo di crittografia
- Criteri opzionali per filtrare gli oggetti per data di creazione, nome chiave, dimensioni, classe di archiviazione e tag

Criteri di filtro degli oggetti

Per filtrare l'elenco degli oggetti da includere in un manifesto generato automaticamente, puoi specificare i seguenti criteri. Per ulteriori informazioni, consulta JobManifestGeneratorFilter nella Documentazione di riferimento delle API Amazon S3.

CreatedAfter

Se fornito, il manifesto generato include solo oggetti del bucket di origine creati dopo questo periodo.

CreatedBefore

Se fornito, il manifesto generato include solo oggetti del bucket di origine creati prima di questo periodo.

EligibleForReplication

Se fornito, il manifesto generato include oggetti solo se sono idonei alla replica in base alla configurazione di replica sul bucket di origine.

KeyNameConstraint

Se fornito, il manifesto generato include solo oggetti bucket di origine le cui chiavi degli oggetti corrispondono ai vincoli di stringa specificati per, e. MatchAnySubstringMatchAnyPrefixMatchAnySuffix

MatchAnySubstring— Se fornito, il manifesto generato include oggetti se la stringa specificata appare in un punto qualsiasi della stringa chiave dell'oggetto.

MatchAnyPrefix— Se fornito, il manifesto generato include oggetti se la stringa specificata appare all'inizio della stringa chiave dell'oggetto.

MatchAnySuffix— Se fornito, il manifesto generato include oggetti se la stringa specificata appare alla fine della stringa chiave dell'oggetto.

MatchAnyStorageClass

Se fornito, il manifesto generato include solo oggetti del bucket di origine archiviati con la classe di archiviazione specificata.

ObjectReplicationStatuses

Se fornito, il manifesto generato include solo oggetti del bucket di origine che dispongono di uno degli stati di replica specificati.

ObjectSizeGreaterThanBytes

Se fornito, il manifesto generato include solo oggetti del bucket di origine la cui dimensione file è maggiore del numero di byte specificato.

ObjectSizeLessThanBytes

Se fornito, il manifesto generato include solo oggetti del bucket di origine la cui dimensione file è minore del numero di byte specificato.

> **Note**
>
> Non è possibile clonare la maggior parte dei processi che hanno generato manifesti automaticamente. I processi di replica batch possono essere clonati, tranne quando utilizzano i criteri di filtro del manifesto `KeyNameConstraint`, `MatchAnyStorageClass`, `ObjectSizeGreaterThanBytes` o `ObjectSizeLessThanBytes`.

La sintassi per specificare i criteri del manifesto varia a seconda del metodo utilizzato per creare il processo. Per alcuni esempi, consulta Creazione di un processo.

Creazione di un processo

Puoi creare job S3 Batch Operations utilizzando la console Amazon S3 AWS CLI AWS , gli SDK o l'API REST di Amazon S3.

Per ulteriori informazioni sulla creazione di una richiesta di processo, consulta la sezione Elementi della richiesta di un processo di operazioni in batch.

Prerequisiti

Prima di creare un processo Operazioni in batch, conferma di aver configurato le autorizzazioni pertinenti. Per ulteriori informazioni, consulta Concessione delle autorizzazioni per le operazioni in batch Amazon S3.

Utilizzo della console S3

Per creare un processo batch

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nella barra di navigazione nella parte superiore della pagina, scegli il nome del file attualmente visualizzato Regione AWS. Quindi, scegli la regione in cui desideri creare il tuo lavoro.

> **Note**
>
> Per le operazioni di copia, è necessario creare il lavoro nella stessa regione del bucket di destinazione. Per tutte le altre operazioni, è necessario creare il lavoro nella stessa regione degli oggetti nel manifesto.

3. Scegli Batch Operations nel riquadro di navigazione a sinistra della console Amazon S3.
4. Scegli Crea processo.
5. Visualizza Regione AWSdove vuoi creare il tuo lavoro.
6. In Formato manifest scegliere il tipo di oggetto manifest da usare.

   - Se si sceglie S3 inventory report (Report di inventario S3), immettere il percorso dell'oggetto manifest.json generato da Amazon S3 come parte del report dell'inventario in formato CSV e, facoltativamente, l'ID versione dell'oggetto manifest se si desidera utilizzare una versione diversa da quella più recente.
   - Se si sceglie CSV, immettere il percorso di un oggetto manifest in formato CSV. L'oggetto manifest deve avere il formato descritto nella console. Facoltativamente, è possibile includere l'ID versione dell'oggetto manifest se si desidera utilizzare una versione diversa da quella più recente.

   > **Note**
   >
   > La console Amazon S3 supporta la generazione manifesto automatica solo per i processi di replica batch. Per tutti gli altri tipi di job, se desideri che Amazon S3 generi automaticamente un manifesto in base ai criteri di filtro specificati, devi configurare il job utilizzando gli AWS SDK o l' AWS CLI API REST di Amazon S3.

7. Seleziona Successivo.
8. In Operation (Operazione) scegliere l'operazione che si desidera eseguire su tutti gli oggetti elencati nel manifest. Inserire le informazioni per l'operazione selezionata, quindi scegliere Next (Avanti).
9. Inserire le informazioni per Configure additional options (Configura opzioni aggiuntive), quindi scegliere Next (Avanti).
10. Per Review (Revisione), verificare le impostazioni. Se è necessario apportare modifiche, scegliere Previous (Precedente). In caso contrario, scegliere Create Job (Crea processo).

Utilizzando il AWS CLI

Specify manifest

Nell'esempio seguente viene illustrato come creare un processo Operazioni in batch S3 `S3PutObjectTagging` che agisce sugli oggetti elencati in un file manifesto esistente.

Per creare un processo **`S3PutObjectTagging`** di operazioni in batch

1. Utilizza i seguenti comandi per creare un ruolo AWS Identity and Access Management (IAM), quindi crea una policy IAM per assegnare le autorizzazioni pertinenti. Il ruolo e la policy seguenti concedono l'autorizzazione Amazon S3 per aggiungere tag degli oggetti, necessari per creare il processo in una fase successiva.

   a. Utilizza il comando di esempio seguente per creare un ruolo IAM utilizzato da Operazioni in batch. Per utilizzare questo comando di esempio, sostituisci *`S3BatchJobRole`* con il nome che desideri assegnare al ruolo.

```
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"
      }
   ]
}'
```

   Registrare l'Amazon Resource Name (ARN) del ruolo. Quando si crea un processo sarà necessario specificare l'ARN.

   b. Utilizza il comando di esempio seguente per creare una policy IAM con le autorizzazioni necessarie e collegala al ruolo IAM creato nella fase precedente. Per ulteriori informazioni sulle autorizzazioni necessarie, consulta Concessione delle autorizzazioni per le operazioni in batch Amazon S3.

> **Note**
>
> I processi Operazioni in batch che eseguono azioni su bucket di directory richiedono autorizzazioni specifiche. Per ulteriori informazioni, consulta AWS Identity and Access Management (IAM) for S3 Express One Zone.

Per utilizzare questo comando di esempio, sostituisci *user input placeholders* come segue:

- Sostituisci *S3BatchJobRole* con il nome del ruolo IAM. Assicurati che questo nome corrisponda al nome utilizzato in precedenza.
- Sostituisci *PutObjectTaggingBatchJobPolicy* con il nome che desideri assegnare alla policy IAM.
- Sostituisci *example-s3-destination-bucket* con il nome del bucket contenente gli oggetti a cui desideri applicare i tag.
- Sostituisci *DOC-EXAMPLE-MANIFEST-BUCKET* con il nome del bucket contenente il manifesto.
- Sostituisci *DOC-EXAMPLE-REPORT-BUCKET* con il nome del bucket a cui desideri venga inviato il report di completamento.

```
aws iam put-role-policy \
  --role-name S3BatchJobRole \
  --policy-name PutObjectTaggingBatchJobPolicy \
  --policy-document '{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObjectTagging",
        "s3:PutObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::example-s3-destination-bucket/*"
    },
    {
      "Effect": "Allow",
```

```
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-BUCKET/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject",
        "s3:GetBucketLocation"
      ],
      "Resource":[
        "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET/*"
      ]
    }
  ]
}'
```

2. Utilizza il comando di esempio seguente per creare un processo `S3PutObjectTagging`.

   Il file `manifest.csv` fornisce un elenco di valori di bucket e chiave di oggetto. Il processo applica i tag specificati agli oggetti identificati nel manifesto. `ETag` è l'ETag dell'oggetto `manifest.csv`, che è possibile ottenere dalla console di Amazon S3. Questa richiesta specifica il parametro `no-confirmation-required`, in modo da poter eseguire il processo senza doverlo confermare con il comando `update-job-status`. Per ulteriori informazioni, consulta la sezione create-job nella Documentazione di riferimento della AWS CLI .

   Per utilizzare questo comando di esempio, sostituisci *`user input placeholders`* con le tue informazioni. Sostituisci *`IAM-role`* con l'ARN del ruolo IAM creato in precedenza.

```
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
```

In risposta, Amazon S3 restituisce un ID processo, ad esempio, `00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c`). L'ID processo è necessario per identificare, monitorare e modificare il processo.

Generate manifest

Nell'esempio seguente viene illustrato come creare un processo Operazioni in batch S3 `S3DeleteObjectTagging` che genera automaticamente un manifesto in base ai criteri di filtro degli oggetti. Questi criteri includono la data di creazione, il nome della chiave, le dimensioni, la classe di archiviazione e i tag.

Per creare un processo **`S3DeleteObjectTagging`** di operazioni in batch

1. Utilizza i seguenti comandi per creare un ruolo AWS Identity and Access Management (IAM), quindi crea una policy IAM per assegnare le autorizzazioni. Il ruolo e la policy seguenti concedono l'autorizzazione Amazon S3 per eliminare tag di oggetti, necessari quando si crea il processo in una fase successiva.

   a. Utilizza il comando di esempio seguente per creare un ruolo IAM utilizzato da Operazioni in batch. Per utilizzare questo comando di esempio, sostituisci *`S3BatchJobRole`* con il nome che desideri assegnare al ruolo.

   ```
   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"
      }
   ]
}'
```

Registrare l'Amazon Resource Name (ARN) del ruolo. Quando si crea un processo sarà necessario specificare l'ARN.

b. Utilizza il comando di esempio seguente per creare una policy IAM con le autorizzazioni necessarie e collegala al ruolo IAM creato nella fase precedente. Per ulteriori informazioni sulle autorizzazioni necessarie, consulta Concessione delle autorizzazioni per le operazioni in batch Amazon S3.

> **Note**
>
> I processi Operazioni in batch che eseguono azioni su bucket di directory richiedono autorizzazioni specifiche. Per ulteriori informazioni, consulta AWS Identity and Access Management (IAM) for S3 Express One Zone.

Per utilizzare questo comando di esempio, sostituisci *user input placeholders* come segue:

- Sostituisci *S3BatchJobRole* con il nome del ruolo IAM. Assicurati che questo nome corrisponda al nome utilizzato in precedenza.
- Sostituisci *DeleteObjectTaggingBatchJobPolicy* con il nome che desideri assegnare alla policy IAM.
- Sostituisci *example-s3-destination-bucket* con il nome del bucket contenente gli oggetti a cui desideri applicare i tag.
- Sostituisci *DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET* con il nome del bucket in cui desideri salvare il manifesto.
- Sostituisci *DOC-EXAMPLE-REPORT-BUCKET* con il nome del bucket a cui desideri venga inviato il report di completamento.

```
aws iam put-role-policy \
  --role-name S3BatchJobRole \
  --policy-name DeleteObjectTaggingBatchJobPolicy \
  --policy-document '{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:DeleteObjectTagging",
        "s3:DeleteObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::example-s3-destination-bucket/*"
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutInventoryConfiguration"
      ],
      "Resource": "arn:aws:s3:::example-s3-destination-bucket"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource":[
        "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET/*",
```

```
        "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET/*"
      ]
    }
  ]
}'
```

2. Utilizza il comando di esempio seguente per creare il processo `S3DeleteObjectTagging`.

   In questo esempio, i valori nella sezione `--report` specificano il bucket, il prefisso, il formato e l'ambito del report del processo che verrà generato. Nella sezione `--manifest-generator` vengono specificate le informazioni sul bucket di origine contenente gli oggetti su cui agirà il processo, informazioni sull'elenco di output del manifesto che verrà generato per il processo e i criteri di filtro per restringere l'ambito degli oggetti da includere nel manifesto in base a data di creazione, vincoli di nome, dimensioni e classe di archiviazione. Il comando specifica, inoltre, la priorità del processo, il ruolo IAM e la Regione AWS.

   Per ulteriori informazioni, consulta la sezione create-job nella Documentazione di riferimento della AWS CLI .

   Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni. Sostituisci *IAM-role* con l'ARN del ruolo IAM creato in precedenza.

```
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
```

In risposta, Amazon S3 restituisce un ID processo, ad esempio, `00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c`). Questo ID processo è necessario per identificare, monitorare e modificare il processo.

Utilizzando il AWS SDK for Java

Specify manifest

Nell'esempio seguente viene illustrato come creare un processo Operazioni in batch S3 `S3PutObjectTagging` che agisce sugli oggetti elencati in un file manifesto esistente. Per utilizzare questo comando, sostituisci *`user input placeholders`* con le tue informazioni.

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

Nell'esempio seguente viene illustrato come creare un processo Operazioni in batch S3 `s3PutObjectCopy` che genera automaticamente un manifesto in base ai criteri di filtro degli oggetti, inclusi data di creazione, nome chiave e dimensioni. Per utilizzare questo comando, sostituisci *`user input placeholders`* con le tue informazioni.

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();
            }
        }
    }
```

Utilizzo di REST API

È possibile utilizzare l'API REST per creare un processo di operazioni in batch. Per ulteriori informazioni, consulta CreateJob in Amazon Simple Storage Service API Reference (Guida di riferimento per l'API di Amazon Simple Storage Service).

Risposte di lavoro

Se la richiesta `CreateJob` ha esito positivo, Amazon S3 restituisce un ID processo. L'ID processo è un identificatore univoco che Amazon S3 genera automaticamente per permetterti di identificare il processo di operazioni in batch e di monitorarne lo stato.

Quando crei un lavoro tramite gli AWS SDK o l' AWS CLI API REST, puoi impostare S3 Batch Operations in modo che inizi a elaborare il lavoro automaticamente. Il processo viene eseguito appena è pronto anziché attendere in coda ad altri processi con priorità più alta.

Quando crei un processo con la console Amazon S3, devi rivedere i dettagli del processo e confermare che desideri eseguirlo prima che venga elaborato da Operazioni in batch. Se un processo rimane nello stato di sospensione per più di 30 giorni, avrà un esito negativo.

Operazioni supportate dalle operazioni in batch S3

Le operazioni in batch S3 supportano diverse operazioni. Negli argomenti di questa sezione vengono descritte tutte queste operazioni.

Copia oggetti

L'operazione Copia copia ogni oggetto specificato nel manifest. È possibile copiare oggetti in un bucket nella stessa regione AWS o in una regione diversa. S3 Batch Operations supporta la maggior parte delle opzioni disponibili tramite Amazon S3 per la copia di oggetti. Queste opzioni includono l'impostazione dei metadati degli oggetti, l'impostazione delle autorizzazioni e la modifica di una classe di storage di un oggetto.

È possibile utilizzare anche l'operazione di copia per copiare gli oggetti non crittografati esistenti e scrivere i nuovi oggetti crittografati nello stesso bucket. Per ulteriori informazioni, consulta Crittografia di oggetti con le operazioni in batch di Amazon S3.

Quando esegui la copia degli oggetti puoi modificare l'algoritmo di checksum utilizzato per calcolare il checksum dell'oggetto. Se gli oggetti non hanno un checksum aggiuntivo calcolato, puoi anche aggiungerne uno specificando l'algoritmo di checksum che Amazon S3 deve utilizzare. Per ulteriori informazioni, consulta Verifica dell'integrità degli oggetti.

Per maggiori informazioni sulla copia di oggetti in Amazon S3, nonché sui parametri obbligatori e facoltativi, consulta Copiare, spostare e rinominare oggetti in questa guida e CopyObject nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Restrizioni e limitazioni

- Tutti gli oggetti di origine devono trovarsi in un bucket.
- Tutti gli oggetti di destinazione devono trovarsi in un bucket.
- È necessario disporre delle autorizzazioni di lettura per il bucket di origine e delle autorizzazioni di scrittura per il bucket di destinazione.

- Gli oggetti da copiare non possono avere dimensioni superiori a 5 GB.
- Se tenti di copiare oggetti dalle classi di archiviazione S3 Glacier Flexible o S3 Glacier Deep Archive nella classe di archiviazione S3 Standard, prima devi ripristinare questi oggetti. Per ulteriori informazioni, consultare Ripristino di un oggetto archiviato.
- I processi di copia devono essere creati nella Regione di destinazione, ovvero nella Regione in cui si intende copiare gli oggetti.
- Sono supportate tutte le opzioni Copy, ad eccezione dei controlli condizionali su ETag e della crittografia lato server con chiavi di crittografia fornite dal cliente.
- Se i bucket sono senza versione, gli oggetti con gli stessi nomi di chiave verranno sovrascritti.
- Gli oggetti non vengono necessariamente copiati nello stesso ordine in cui appaiono nel manifest. Per i bucket con versione, se è importante mantenere l'ordine di versione corrente/non corrente, è necessario copiare prima tutte le versioni non correnti. Quindi, al termine del primo processo, copia le versioni correnti in un processo successivo.
- La copia di oggetti nella classe Reduced Redundancy Storage (RRS) non è supportata.

Copia di oggetti mediante operazioni in batch S3

È possibile usare le operazioni in batch S3 per creare un processo di copia PUT per copiare gli oggetti nello stesso account o in un diverso account di destinazione. Le sezioni seguenti contengono esempi di come memorizzare e utilizzare un manifest che si trova in un account diverso. Nella prima sezione è possibile usare l'inventario Amazon S3 per consegnare il report di inventario all'account di destinazione e utilizzarlo durante la creazione del processo, oppure usare un manifest con valori separati da virgole (CSV) nell'account di origine o di destinazione come illustrato nel secondo esempio. Nel terzo esempio viene illustrato come utilizzare l'operazione Copia per attivare la crittografia con chiave bucket S3 sugli oggetti esistenti.

Esempi di operazioni di copia

- Utilizzo di un report di inventario consegnato all'account di destinazione per copiare oggetti tra Account AWS
- Utilizzo di un manifest CSV archiviato nell'account di origine per copiare oggetti tra Account AWS
- Utilizzo delle operazioni in batch S3 per crittografare oggetti con chiavi bucket S3

Utilizzo di un report di inventario consegnato all'account di destinazione per copiare oggetti tra Account AWS

Utilizza l'inventario Amazon S3 per creare un report dell'inventario e utilizza questo report per creare un elenco di oggetti da copiare mediante operazioni in batch S3. Per maggiori informazioni sull'utilizzo di un manifest CSV nell'account di origine o di destinazione, consulta the section called “Utilizzo di un manifest CSV per copiare oggetti tra Account AWS”.

L'inventario Amazon S3 genera inventari degli oggetti in un bucket. L’elenco risultante viene pubblicato in un file di output. Il bucket inventariato è chiamato bucket di origine, mentre il bucket dove viene memorizzato il file del rapporto di inventario è chiamato bucket di destinazione.

Il report di Amazon S3 Inventory può essere configurato per essere consegnato a un altro Account AWS. Questo consente alle operazioni in batch S3 di leggere il report di inventario quando il processo viene creato nell'account di destinazione.

Per ulteriori informazioni sui bucket di origine e destinazione degli inventari Amazon S3, consulta Bucket di origine e di destinazione.

Il modo più semplice per impostare un inventario è quello di utilizzare la AWS Management Console; tuttavia è possibile utilizzare l'API REST, AWS Command Line Interface (AWS CLI) o gli SDK AWS.

La procedura della console seguente contiene le fasi di livello elevato per la configurazione delle autorizzazioni per un processo di operazioni in batch S3. In questa procedura, si copiano gli oggetti da un account di origine a un account di destinazione, con il report di inventario archiviato nell'account di destinazione.

Per configurare Amazon S3 Inventory per bucket di origine e di destinazione di proprietà di account diversi

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Scegliere un bucket di destinazione in cui memorizzare il rapporto di inventario.

   Decidere un bucket manifest di destinazione per memorizzare il rapporto di inventario. In questa procedura, l’account di destinazione è l’account che possiede sia il bucket manifest di destinazione sia il bucket in cui vengono copiati gli oggetti.
3. Configurare un inventario per elencare gli oggetti in un bucket di origine e pubblicare l'elenco sul bucket manifest di destinazione.

   Configurare un elenco di inventario per un bucket di origine. In questa fase, specificare il bucket di destinazione in cui si desidera memorizzare l’elenco. Il rapporto di inventario per il bucket di

origine viene pubblicato nel bucket di destinazione. In questa procedura, l'account di origine è l'account che possiede il bucket di origine.

Per informazioni su come utilizzare la console per configurare un inventario o su come crittografare il file dell'elenco inventario, consulta Configurazione di Amazon S3 Inventory.

Scegliere CSV come formato di output.

Quando si inseriscono le informazioni per il bucket di destinazione, scegliere Buckets in another account (Bucket in un altro account). Quindi inserire il nome del bucket manifest di destinazione. Facoltativamente, è possibile inserire l'ID account dell'account di destinazione.

Una volta salvata la configurazione dell'inventario, la console visualizza un messaggio simile al seguente:

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.

La console visualizza quindi una policy di bucket che può essere usata per il bucket di destinazione.

4. Copiare la policy del bucket di destinazione visualizzata sulla console.
5. Nell'account di destinazione, aggiungere la policy di bucket copiata nel bucket manifest di destinazione in cui è memorizzato il rapporto di inventario.
6. Creare un ruolo nell'account di destinazione basato sulla policy di attendibilità delle operazioni in batch S3. Per ulteriori informazioni sulla policy di attendibilità, consultare Policy di trust.

   Per ulteriori informazioni sulla creazione di un ruolo, consulta Creazione di un ruolo per delegare le autorizzazioni a un servizio AWS nella Guida per l'utente di IAM.

   Immettere un nome per il ruolo (il ruolo di esempio usa il nome `BatchOperationsDestinationRoleCOPY`). Scegliere il servizio S3, quindi scegliere il caso d'uso S3 bucket Batch Operations, che applica la policy di attendibilità al ruolo.

   Quindi scegliere Create policy (Crea policy) per collegare la seguente policy al ruolo.

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

Il ruolo usa la policy per concedere l'autorizzazione `batchoperations.s3.amazonaws.com` per leggere il manifest nel bucket di destinazione. Concede anche le autorizzazioni per il GET di oggetti, liste di controllo accessi (ACL), tag e versioni nel bucket di oggetti di origine. Inoltre, concede le autorizzazioni per il PUT di oggetti, liste di controllo accessi (ACL), tag e versioni nel bucket di oggetti di destinazione.

7. Nell'account di origine, creare una policy di bucket per il bucket di origine che assegna il ruolo creato nella fase precedente per il GET di oggetti, liste di controllo accessi (ACL), tag e versioni nel bucket di origine. Questa fase consente alle operazioni in batch S3 di ottenere oggetti dal bucket di origine tramite il ruolo trusted.

   Segue un esempio della policy di bucket per l'account di origine.

```
{
  "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. Quando il report di inventario è disponibile, creare un processo PUT object copy delle operazioni in batch S3 nell'account di destinazione, scegliendo il report di inventario dal bucket manifest di destinazione. È necessario l'ARN del ruolo creato nell'account di destinazione.

   Per informazioni generali sulla creazione di un processo, consultare Creazione di un processo di operazioni in batch S3.

   Per informazioni sulla creazione di un processo mediante la console, consulta Creazione di un processo di operazioni in batch S3.

Utilizzo di un manifest CSV archiviato nell'account di origine per copiare oggetti tra Account AWS

È possibile usare un file CSV archiviato in un altro Account AWS come manifest per un processo di operazioni in batch S3. Per l'utilizzo di un report di inventario S3, consulta the section called "Utilizzo di un report di inventario per copiare oggetti tra Account AWS".

Nella procedura seguente viene illustrato come configurare le autorizzazioni quando si utilizza un processo di operazioni in batch S3 per copiare oggetti da un account di origine a un account di destinazione con il file manifest CSV archiviato nell'account di origine.

Come configurare un manifest CSV archiviato in un altro Account AWS

1. Creare un ruolo nell'account di destinazione basato sulla policy di attendibilità delle operazioni in batch S3. In questa procedura, l'account di destinazione è l'account in cui vengono copiati gli oggetti.

   Per ulteriori informazioni sulla policy di attendibilità, consultare Policy di trust.

   Per ulteriori informazioni sulla creazione di un ruolo, consulta Creazione di un ruolo per delegare le autorizzazioni a un servizio AWS nella Guida per l'utente di IAM.

   Se si crea il ruolo usando la console, immettere un nome per il ruolo (il ruolo di esempio usa il nome `BatchOperationsDestinationRoleCOPY`). Scegliere il servizio S3, quindi scegliere il caso d'uso S3 bucket Batch Operations, che applica la policy di attendibilità al ruolo.

   Quindi scegliere Create policy (Crea policy) per collegare la seguente policy al ruolo.

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

```
  ]
}
```

Usando la policy, il ruolo concede l'autorizzazione `batchoperations.s3.amazonaws.com` per leggere il manifest nel bucket manifest di origine. Concede le autorizzazioni per il GET di oggetti, liste di controllo accessi (ACL), tag e versioni nel bucket di oggetti di origine. Inoltre, concede le autorizzazioni per il PUT di oggetti, liste di controllo accessi (ACL), tag e versioni nel bucket di oggetti di destinazione.

2. Nell'account di origine, creare una policy di bucket per il bucket che contiene il manifest per assegnare il ruolo creato nella fase precedente per il GET di oggetti e versioni nel bucket manifest di origine.

   Questa fase consente alle operazioni in batch S3 di leggere il manifest usando il ruolo trusted. Applicare la policy di bucket al bucket che contiene il manifest.

   Segue un esempio della policy di bucket da applicare al bucket manifest di origine.

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

Questa policy inoltre concede le autorizzazioni per garantire a un utente della console che sta creando un processo nell'account di destinazione le stesse autorizzazioni nel bucket manifest di origine tramite la stessa policy di bucket.

3. Nell'account di origine, creare una policy del bucket per il bucket di origine che concede il ruolo creato al GET di oggetti, alle liste di controllo accessi (ACL), ai tag e alle versioni nel bucket dell'oggetto di origine. Le operazioni in batch S3 possono quindi ottenere oggetti dal bucket di origine tramite il ruolo trusted.

   Segue un esempio della policy di bucket per il bucket che contiene gli oggetti di origine.

```
{
  "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. Creare un processo di operazioni in batch S3 nell'account di destinazione. È necessario l'ARN del ruolo creato nell'account di destinazione.

   Per informazioni generali sulla creazione di un processo, consultare Creazione di un processo di operazioni in batch S3.

   Per informazioni sulla creazione di un processo mediante la console, consulta Creazione di un processo di operazioni in batch S3.

Utilizzo delle operazioni in batch S3 per crittografare oggetti con chiavi bucket S3

In questa sezione, puoi utilizzare l'operazione Copia delle operazioni in batch Amazon S3 per identificare e attivare la crittografia delle chiavi bucket S3 sugli oggetti esistenti. Per ulteriori informazioni sulle chiavi bucket S3, consulta Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3 e Configurazione del bucket per utilizzare una chiave bucket S3 con SSE-KMS per nuovi oggetti.

Gli argomenti trattati in questo esempio sono i seguenti:

Argomenti

Prerequisiti

Per seguire la procedura descritta in questa sezione, è necessario un Account AWS e almeno un bucket S3 che contenga i file di lavoro e i risultati crittografati. È inoltre possibile trovare utile buona parte della documentazione esistente delle operazioni in batch S3, inclusi i seguenti argomenti:

- Nozioni di base sulle operazioni in batch S3
- Creazione di un processo di operazioni in batch S3
- Operazioni supportate dalle operazioni in batch S3
- Gestione dei processi di operazioni in batch Amazon S3

Fase 1: Ottenimento dell'elenco di oggetti tramite Amazon S3 Inventory

Per iniziare, identifica il bucket S3 che contiene gli oggetti da crittografare e recupera un elenco del suo contenuto. Un report di inventario di Amazon S3 è il modo più conveniente per farlo. Il report fornisce l'elenco degli oggetti in un bucket insieme ai metadati associati. Bucket di origine si riferisce al bucket inventariato mentre bucket di destinazione si riferisce al bucket in cui viene archiviato il file del report di inventario. Per ulteriori informazioni sui bucket di origine e destinazione degli inventari Amazon S3, consulta Amazon S3 Inventory.

Il modo più semplice per configurare un inventario consiste nell'utilizzare la AWS Management Console. Puoi utilizzare anche l'API REST, AWS Command Line Interface (AWS CLI) o gli SDK AWS. Prima di completare questa procedura, accedi alla console e apri la console di Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/. Se si verificano errori di autorizzazione negata, aggiungi una policy bucket al bucket di destinazione. Per ulteriori informazioni, consulta Concedere autorizzazioni per S3 Inventory e S3 Analytics.

Come ottenere un elenco di oggetti tramite l'inventario S3

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione seleziona Bucket e scegli un bucket che contiene oggetti da crittografare.
3. Nella scheda Gestione, passa alla sezione Configurazioni di inventario e seleziona Crea configurazione di inventario.
4. Assegna un nome al tuo nuovo inventario, inserisci il nome del bucket S3 di destinazione e, facoltativamente, crea un prefisso di destinazione per Amazon S3 per assegnare gli oggetti in quel bucket.
5. Per Formato di output, seleziona CSV.
6. (Facoltativo) Nella sezione Campi aggiuntivi - facoltativi, seleziona Crittografia e tutti gli altri campi del report che ti interessano. Imposta la frequenza per le consegne del report su Giornaliero in modo che il primo report venga consegnato al bucket quanto prima.
7. Seleziona Crea per salvare la configurazione.

Amazon S3 può richiedere fino a 48 ore per consegnare il primo report, quindi controlla quando arriva. Dopo aver ricevuto il primo report, passa alla sezione successiva per filtrare il contenuto del report di S3 Inventory. Se non desideri più ricevere report di inventario per questo bucket, elimina la configurazione dell'inventario S3. In caso contrario, S3 fornisce report su una pianificazione giornaliera o settimanale.

Un elenco di inventario non è una point-in-time visualizzazione singola di tutti gli oggetti. Gli elenchi di inventario sono uno snapshot in sequenza di voci del bucket, che sono a consistenza finale (cioè, l'elenco potrebbe non includere oggetti aggiunti o eliminati di recente). La combinazione di inventario S3 e operazioni in batch S3 funziona meglio quando si lavora con oggetti statici o con un set di oggetti creato due o più giorni prima. Per lavorare con dati più recenti, utilizza l'operazione API ListObjectsV2 (GET Bucket) per creare manualmente l'elenco di oggetti. Se necessario, ripeti la

procedura per i giorni successivi o fino a quando il report di inventario non mostra lo stato desiderato per tutte le chiavi.

Fase 2: Filtro dell'elenco degli oggetti con S3 Select

Dopo aver ricevuto il report di S3 Inventory, puoi filtrare il contenuto del report in modo da visualizzare solo gli oggetti non crittografati con le chiavi del bucket S3. Se desideri crittografare tutti gli oggetti del bucket con le chiavi bucket S3, puoi ignorare questo passaggio. Tuttavia, filtrare il report di inventario S3 in questa fase consente di risparmiare tempo e spese per la ricrittografia degli oggetti precedentemente crittografati.

Sebbene i passaggi seguenti illustrino come filtrare utilizzando Amazon S3 Select puoi utilizzare anche Amazon Athena. Per decidere quale strumento utilizzare, dai un'occhiata al file `manifest.json` del report di inventario S3. Questo file riporta il numero di file di dati associati a tale report. Se il numero è grande, usa Amazon Athena perché viene eseguito su più oggetti S3, mentre S3 Select funziona su un oggetto alla volta. Per maggiori informazioni sull'utilizzo di Amazon S3 e Athena insieme, consulta Esecuzione di query sull'inventario Amazon S3 con Amazon Athena e Uso di Athena nel post del blogCrittografia di oggetti con operazioni in batch di Amazon S3.

Come filtrare il report di inventario S3 tramite S3 Select

1. Apri il file `manifest.json` dal report di inventario e guarda la sezione `fileSchema` del JSON. Questa sezione informa la query che si esegue sui dati.

   Il seguente JSON è un file `manifest.json` di esempio per un inventario in formato CSV su un bucket con il controllo delle versioni abilitato. A seconda di come hai configurato il report di inventario, il manifest potrebbe apparire diverso.

```
  {
    "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"
      }
    ]
}
```

Se il controllo delle versioni non è attivato nel bucket o se hai deciso di eseguire il report per le versioni più recenti, la `fileSchema` è `Bucket`, `Key` e `BucketKeyStatus`.

Se il controllo delle versioni è attivato, a seconda di come è stato impostato il report di inventario, il `fileSchema` può includere quanto segue: `Bucket`, `Key`, `VersionId`, `IsLatest`, `IsDeleteMarker`, `BucketKeyStatus`. Quindi presta attenzione alle colonne 1, 2, 3 e 6 quando esegui la query.

Per eseguire il processo, la funzionalità Operazioni in batch Amazon S3 richiede come input il bucket, la chiave e l'ID versione, oltre al campo in base al quale eseguire la ricerca, ovvero `BucketKeyStatus`. Il campo ID versione non è necessario ma può risultare utile specificarlo se si utilizza un bucket con controllo delle versioni. Per ulteriori informazioni, consulta Utilizzo di oggetti in un bucket che supporta la funzione Controllo delle versioni.

2. Individua i file di dati per il report di inventario. L'oggetto `manifest.json` riporta i file di dati in files.
3. Dopo aver individuato e selezionato il file di dati nella console S3, seleziona Operazioni quindi scegli Query con S3 Select.
4. Mantieni i valori preimpostati per CSV, virgola e GZIP e seleziona Successivo.
5. Per rivedere il formato del report dell'inventario prima di procedere, scegli Mostra anteprima file.
6. Immetti le colonne a cui fare riferimento nel campo SQL expression (Espressione SQL), quindi seleziona Run SQL (Esegui SQL). L'espressione seguente restituisce le colonne 1-3 per tutti gli oggetti senza chiave bucket S3 configurata.

   ```
   select s._1, s._2, s._3 from s3object s where s._6 = 'DISABLED'
   ```

   Di seguito sono riportati i risultati di esempio.

   ```
   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. Scarica i risultati, salvali in un formato CSV e caricali in Amazon S3 come elenco di oggetti per il processo di operazioni in batch S3.
8. Se disponi di più file manifest, esegui Query con S3 Select anche su quelli. A seconda delle dimensioni dei risultati, è possibile combinare gli elenchi ed eseguire un singolo processo di operazioni in batch S3 oppure eseguire ogni elenco come processo separato.

   Considera il prezzo dell'esecuzione di ciascun processo di operazioni in batch S3 quando decidi il numero di processi da eseguire.

Fase 3: Impostazione ed esecuzione del processo di operazioni in batch S3

Ora che disponi di elenchi CSV filtrati di oggetti S3, puoi avviare il processo di operazioni in batch S3 per crittografare gli oggetti con chiavi bucket S3.

Un processo fa riferimento collettivamente all'elenco (manifest) degli oggetti forniti, all'operazione eseguita e ai parametri specificati. Il modo più semplice per crittografare questo set di oggetti consiste nell'utilizzare l'operazione di copia di PUT e specificare lo stesso prefisso di destinazione degli oggetti elencati nel manifest. Ciò sovrascrive gli oggetti esistenti in un bucket senza versione o, con il controllo delle versioni attivato, crea una versione più recente e crittografata degli oggetti.

Come parte della copia degli oggetti, specifica che Amazon S3 deve crittografare l'oggetto con la crittografia SSE-KMS e S3. Questo processo copia gli oggetti in modo che tutti gli oggetti visualizzino una data di creazione aggiornata al completamento, indipendentemente dal momento in cui sono stati aggiunti originariamente a S3. Specifica inoltre le altre proprietà per l'insieme di oggetti come parte del processo di operazioni in batch S3, inclusi i tag oggetto e la classe di archiviazione.

Fasi secondarie

- Impostazione della policy IAM
- Impostazione del ruolo IAM nelle operazioni in batch
- Attivazione delle chiavi bucket S3 per un bucket esistente
- Creazione di un processo di operazioni in batch S3

- Esecuzione del processo di operazioni in batch
- Da sapere

Impostazione della policy IAM

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy e Crea policy.
3. Selezionare la scheda JSON. Seleziona Modifica policy e aggiungi la policy IAM di esempio visualizzata nel seguente blocco di codice.

   Dopo aver copiato l'esempio di policy nella tua console IAM, sostituisci quanto segue:

   a. Sostituisci *SOURCE_BUCKET_FOR_COPY* con il nome del tuo bucket di origine.
   b. Sostituisci *DESTINATION_BUCKET_FOR_COPY* con il nome del tuo bucket di destinazione.
   c. Sostituisci *MANIFEST_KEY* con il nome del tuo oggetto manifesto.
   d. Sostituisci *REPORT_BUCKET* con il nome del bucket in cui desideri salvare i report.

```
{
  "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. Scegliere Next: Tags (Successivo: Tag).
5. Aggiungi tutti i tag desiderati (facoltativo) e seleziona Successivo: Rivedi.
6. Aggiungi un nome per la policy, facoltativamente, una descrizione quindi scegli Crea policy.
7. Scegli Esamina policy e quindi Salva modifiche.
8. Una volta completata la policy per le operazioni in batch S3, la console ritorna alla pagina Policy di IAM. Filtra in base al nome della policy, scegli il pulsante a sinistra del nome della policy, seleziona Operazioni di policy e scegli Collega.

   Per associare la nuova policy creata a un ruolo IAM, seleziona gli utenti, i gruppi o i ruoli appropriati nell'account e scegli Collega policy. In questo modo si ritorna alla console IAM.

Impostazione del ruolo IAM nelle operazioni in batch

1. Nella console IAM, nel riquadro di navigazione, scegli Ruoli, quindi scegli Crea ruolo.
2. Seleziona Servizio AWS, S3 e Operazioni in batch S3. Quindi scegliere Next: Permissions (Successivo: Autorizzazioni).

3. Inizia a inserire il nome della policy IAM che hai appena creato. Seleziona la casella di controllo in base al nome della policy quando viene visualizzata e scegli Successivo: Tag.
4. (Facoltativo) Aggiungi tag o mantieni vuoti i campi di chiave e valore per questo esercizio. Scegliere Next:Review (Successivo:Rivedi).
5. Specifica un nome per il ruolo e accetta la descrizione predefinita o aggiungine una personalizzata. Seleziona Create role (Crea ruolo).
6. Assicurati che l'utente che crea il processo disponga delle autorizzazioni riportate nell'esempio seguente.

   Sostituisci *{ACCOUNT-ID}* con il tuo ID Account AWS e *{IAM_ROLE_NAME}* con il nome che prevedi di applicare al ruolo IAM che verrà creato nel passaggio della creazione del processo Operazioni in batch più avanti. Per ulteriori informazioni, consulta Concessione delle autorizzazioni per le operazioni in batch Amazon S3.

   ```
   {
   "Sid": "AddIamPermissions",
   "Effect": "Allow",
   "Action": [
   "iam:GetRole",
   "iam:PassRole"
   ],
   "Resource": "arn:aws:iam::ACCOUNT-ID:role/IAM_ROLE_NAME"
   }
   ```

Attivazione delle chiavi bucket S3 per un bucket esistente

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket scegli il bucket per cui desideri abilitare una chiave bucket S3.
3. Scegliere Properties (Proprietà).
4. In Default encryption (Crittografia di default), scegliere Edit (Modifica).
5. In Tipo di crittografia, scegli Chiavi gestite da Amazon S3 (SSE-S3) e Chiave AWS Key Management Service (SSE-KMS).
6. Se hai scelto Chiave AWS Key Management Service (SSE-KMS), in AWS KMS key puoi specificare la chiave AWS KMS tramite una delle seguenti opzioni.

   - Per effettuare una selezione in un elenco di chiavi KMS disponibili, seleziona Scegli tra le chiavi AWS KMS. Nell'elenco di chiavi disponibili, scegli una chiave KMS di crittografia

simmetrica nella stessa regione del bucket. Nell'elenco vengono visualizzate sia la chiave gestita da AWS (`aws/s3`) che le chiavi gestite dal cliente.

- Per inserire l'ARN della chiave KMS, scegli Inserisci l'ARN della AWS KMS, quindi inserisci l'ARN della chiave KMS nel campo visualizzato.
- Per creare una chiave gestita dal cliente nella console AWS KMS, scegli Crea una chiave KMS.

7. In Chiave bucket, seleziona Abilita quindi scegli Salva modifiche.

Ora che la chiave del bucket S3 è attivata a livello di bucket, gli oggetti caricati, modificati o copiati in questo bucket erediteranno questa configurazione di crittografia per impostazione predefinita. Sono inclusi anche gli oggetti copiati tramite la funzionalità Operazioni in batch Amazon S3.

Creazione di un processo di operazioni in batch S3

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione seleziona Operazioni in batch, quindi scegli Crea processo.
3. Seleziona la regione in cui si trovano i tuoi oggetti e scegli CSV come tipo manifest.
4. Specifica il percorso o passare al file manifest CSV creato in precedenza dai risultati di S3 Select (o Athena). Se il manifest contiene ID versione, seleziona tale casella. Seleziona Avanti.
5. Seleziona Copia e scegli il bucket di destinazione della copia. Puoi mantenere la crittografia lato server disattivata. Finché nella destinazione del bucket sono abilitate le chiavi bucket S3, l'operazione di copia applica le chiavi bucket S3 al bucket di destinazione.
6. (Facoltativo) Scegli una classe di archiviazione e gli altri parametri come desiderato. I parametri specificati in questo passaggio si applicano a tutte le operazioni eseguite sugli oggetti riportati nel manifest. Seleziona Avanti.
7. Per configurare la crittografia lato server, completa la seguente procedura:

    a. In Crittografia lato server completa la seguente procedura:

    - Per conservare le impostazioni relative ai bucket per la crittografia predefinita degli oggetti lato server durante l'archiviazione in Amazon S3, scegli Non specificare una chiave di crittografia. Finché nella destinazione del bucket sono abilitate le chiavi bucket S3, l'operazione di copia applica la chiave bucket S3 al bucket di destinazione.

> **Note**
>
> Se la policy di bucket per la destinazione specificata richiede la crittografia degli oggetti prima di archiviarli in Amazon S3, è necessario specificare una chiave di crittografia. In caso contrario, la copia degli oggetti nella destinazione avrà esito negativo.

- Per crittografare gli oggetti prima di archiviarli in Amazon S3, scegli Specifica una chiave di crittografia.

b. In Impostazioni di crittografia, se scegli Specifica una chiave di crittografia, devi scegliere Usa le impostazioni del bucket di destinazione per la crittografia predefinita o Ignora le impostazioni del bucket di destinazione per la crittografia predefinita.

c. Se scegli Ignora le impostazioni del bucket di destinazione per la crittografia predefinita, dovrai configurare le seguenti impostazioni di crittografia.

   i. In Tipo di crittografia, scegli Chiavi gestite da Amazon S3 (SSE-S3) o Chiave AWS Key Management Service (SSE-KMS). Per crittografare gli oggetti, SSE-S3 utilizza una delle cifrature di blocco più complesse, lo standard di crittografia avanzata a 256 bit (AES-256). SSE-KMS garantisce un maggiore controllo sulla chiave. Per ulteriori informazioni, consultare Uso della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) e Utilizzo della crittografia lato server con chiavi (SSE-KMS) AWS KMS.

   ii. Se scegli Chiave AWS Key Management Service (SSE-KMS), in AWS KMS key puoi specificare la tua chiave AWS KMS key tramite una delle seguenti opzioni.

   - Per effettuare una selezione in un elenco di chiavi KMS disponibili, seleziona Scegli dalle AWS KMS keys, quindi seleziona una chiave KMS di crittografia simmetrica nella stessa regione del bucket. Nell'elenco vengono visualizzate sia la chiave gestita da AWS (`aws/s3`) che le chiavi gestite dal cliente.
   - Per inserire l'ARN della chiave KMS, scegli Inserisci l'ARN della AWS KMS e quindi specifica l'ARN della chiave KMS nel campo visualizzato.
   - Per creare una chiave gestita dal cliente nella console AWS KMS, scegli Crea una chiave KMS.

   iii. In Chiave bucket scegli Abilita. L'operazione di copia applica una chiave bucket S3 al bucket di destinazione.

8. Assegna al processo una descrizione (o mantieni quella predefinita), impostane il livello di priorità, scegli un tipo di report e specifica il Percorso di destinazione del report di completamento.
9. Nella sezione Autorizzazioni, assicurati di scegliere il ruolo IAM delle operazioni in batch definito in precedenza. Seleziona Avanti.
10. In Rivedi, verificare le impostazioni. Se è necessario apportare modifiche, seleziona Precedente. Dopo aver confermato le impostazioni delle operazioni in batch, seleziona Crea processo.

    Per ulteriori informazioni, consulta Creazione di un processo di operazioni in batch S3.

Esecuzione del processo di operazioni in batch

La procedura guidata di configurazione ti riporta automaticamente alla sezione Operazioni in batch S3 della console di Amazon S3. Le transizioni del tuo nuovo processo dallo stato Nuovoallo stato Preparazione in corso indicano come inizia il processo S3. Durante lo stato Preparazione in corso, S3 legge il manifest del processo, controlla la presenza di errori e calcola il numero di oggetti.

1. Scegli il pulsante Aggiorna nella console di Amazon S3 per verificare lo stato di avanzamento. A seconda delle dimensioni del manifest, la lettura può richiedere minuti o ore.
2. Dopo che S3 ha terminato la lettura del manifest del processo, il processo passa allo stato In attesa di conferma. Scegli il pulsante di opzione a sinistra dell'ID processo, quindi seleziona Esegui processo.
3. Controlla le impostazioni del processo e scegli Esegui processo nell'angolo in basso a destra.

   Dopo l'inizio dell'esecuzione del processo, puoi scegliere il pulsante di aggiornamento per verificarne l'avanzamento tramite la vista del pannello di controllo della console o selezionando il processo specifico.
4. Una volta completato il processo, puoi visualizzare il numero di oggetti con stato Riuscito e Non riuscito per confermare che tutto è stato eseguito come previsto. Se hai abilitato i report del processo, controlla la causa esatta di eventuali operazioni non riuscite nel report.

   Puoi eseguire questa procedura anche utilizzando la AWS CLI, gli SDK AWS o la REST API di Amazon S3. Per ulteriori informazioni sul monitoraggio dello stato del processo e dei report sul completamento, consulta Monitoraggio dei rapporti sullo stato e sul completamento dei processi.

Da sapere

Prendi in considerazione i seguenti problemi quando utilizzi Operazioni in batch Amazon S3 per crittografare gli oggetti con le chiavi bucket S3:

- Verranno addebitati i costi di processi, oggetti e richieste associati alla funzione Operazioni in batch Amazon S3, oltre ai costi associati all'operazione eseguita dalla funzione Operazioni in batch Amazon S3 per tuo conto, inclusi trasferimenti dati, richieste e altri addebiti. Per ulteriori informazioni, consulta Prezzi di Amazon S3.
- Se utilizzi un bucket con versioni, ogni processo di operazioni in batch S3 eseguito crea nuove versioni crittografate degli oggetti. Conserva anche le versioni precedenti configurate senza chiave di bucket S3. Per eliminare le versioni precedenti, imposta una policy di scadenza del ciclo di vita S3 per le versioni non correnti, come descritto in Elementi della configurazione del ciclo di vita.
- L'operazione di copia crea nuovi oggetti con nuove date di creazione, che possono influire sulle operazioni del ciclo di vita, come ad esempio l'archiviazione. Se copi tutti gli oggetti nel bucket, tutte le nuove copie avranno date di creazione identiche o simili. Per identificare ulteriormente questi oggetti e creare regole del ciclo di vita diverse per vari sottoinsiemi di dati, è consigliabile utilizzare i tag oggetto.

Riepilogo

In questa sezione sono stati ordinati gli oggetti esistenti per filtrare i dati già crittografati. Quindi è stata applicata la funzione Chiave bucket S3 su oggetti non crittografati utilizzando la funzionalità Operazioni in batch Amazon S3 per copiare i dati esistenti in un bucket con chiave bucket S3 attivata. Questo processo consente di risparmiare tempo e denaro e al contempo di completare operazioni come la crittografia di tutti gli oggetti esistenti.

Per ulteriori informazioni sulle operazioni in batch, consulta Esecuzione di operazioni in batch su larga scala su oggetti Amazon S3.

Per esempi che mostrano l'operazione di copia con tag mediante la AWS CLI e AWS SDK for Java, consulta la sezione Creazione di un processo Batch Operations con tag di processo utilizzati per l'etichettatura.

Funzione Invoke AWS Lambda

La AWS Lambda funzione Invoke avvia AWS Lambda funzioni per eseguire azioni personalizzate sugli oggetti elencati in un manifesto. Questa sezione descrive come creare una funzione Lambda

da utilizzare con le operazioni in batch Amazon S3 e come creare un processo per richiamare la funzione. Il processo di S3 Batch Operations utilizza l'operazione `LambdaInvoke` per eseguire una funzione Lambda su ogni oggetto elencato in un manifest.

Puoi lavorare con S3 Batch Operations for Lambda utilizzando AWS Management Console AWS Command Line Interface ,AWS CLI() AWS , SDK o API REST. Per ulteriori informazioni sull'utilizzo di Lambda, consulta Nozioni di base su AWS Lambda nella Guida per Developer di AWS Lambda .

Le sezioni seguenti spiegano come iniziare a utilizzare le operazioni in batch S3 con Lambda.

Argomenti

- Utilizzo di Lambda con le operazioni in batch Amazon S3
- Creazione di una funzione Lambda da utilizzare con le operazioni in batch S3
- Creazione di un processo di operazioni in batch Amazon S3 che richiama una funzione Lambda
- Aggiunta di informazioni a livello di attività nei manifest Lambda
- Informazioni sul tutorial su Operazioni in batch Amazon S3

Utilizzo di Lambda con le operazioni in batch Amazon S3

Quando si utilizza S3 Batch Operations con AWS Lambda, è necessario creare nuove funzioni Lambda specifiche per l'uso con S3 Batch Operations. Non puoi riutilizzare funzioni basate su eventi Amazon S3 esistenti con le operazioni in batch S3. Le funzioni evento possono solo ricevere messaggi, non possono restituirli. Le funzioni Lambda utilizzate con le operazioni in batch S3 devono accettare e restituire messaggi. Per ulteriori informazioni sull'uso di Lambda con gli eventi Amazon S3, consulta Using with AWS Lambda Amazon S3 nella Developer Guide.AWS Lambda

Devi creare un processo di operazioni in batch Amazon S3 che richiama la funzione Lambda. Il processo esegue la stessa funzione Lambda su tutti gli oggetti elencati nel manifest. Puoi controllare quali versioni della funzione Lambda utilizzare durante l'elaborazione degli oggetti nel manifest. Le operazioni in batch S3 supportano Amazon Resource Name (ARN) non qualificati, alias e versioni specifiche. Per ulteriori informazioni, consulta Introduzione al controllo delle versioni di AWS Lambda nella Guida per Developer di AWS Lambda .

Se fornisci il processo di operazioni in batch Amazon S3 con una funzione ARN che utilizza un alias o il qualificatore `$LATEST` e aggiorni la versione cui questi puntano, le operazioni in batch S3 iniziano a chiamare la nuova versione della funzione Lambda. Ciò può essere utile quando desideri aggiornare la parte di funzionalità durante un processo di grandi dimensioni. Se non vuoi che le operazioni in

batch S3 modifichino la versione utilizzata, fornisci la versione specifica nel parametro `FunctionARN` durante la creazione del processo.

Utilizzo di Lambda e Operazioni in batch Amazon S3 con bucket di directory

I bucket di directory sono un tipo di bucket Amazon S3 progettato per carichi di lavoro o applicazioni critiche per le prestazioni che richiedono una latenza costante di pochi millisecondi. Per ulteriori informazioni, consulta Directory buckets.

Esistono requisiti speciali per l'utilizzo di Operazioni in batch Amazon S3 per richiamare funzioni Lambda che agiscono su bucket di directory. Ad esempio, è necessario strutturare la richiesta Lambda utilizzando uno schema JSON aggiornato e specificare `InvocationSchemaVersion` **`2.0`** quando si crea il processo. Questo schema aggiornato consente di specificare coppie chiave-valore opzionali per `UserArguments`, che puoi utilizzare per modificare determinati parametri delle funzioni Lambda esistenti. Per ulteriori informazioni, consulta Automatizzare l'elaborazione degli oggetti nei bucket di directory Amazon S3 con S3 Batch Operations AWS Lambda e nel blog sullo storage.AWS

Codici di risposta e dei risultati

S3 Batch Operations richiama la funzione Lambda con uno o più tasti, a ognuno dei quali è associato un tasto. `TaskID` S3 Batch Operations prevede un codice risultato per chiave dalle funzioni Lambda. A tutti gli ID di attività inviati nella richiesta che non vengono restituiti con un codice risultato per chiave verrà assegnato il codice risultante dal campo. `treatMissingKeysAs` `treatMissingKeysAs`è un campo di richiesta opzionale e il valore predefinito è. `TemporaryFailure` La tabella seguente contiene gli altri codici e valori di risultato possibili per il `treatMissingKeysAs` campo.

Codice di risposta	Descrizione
`Succeeded`	L'attività si è conclusa normalmente. Se hai richiesto un rapporto di completamento del processo, la stringa di risultato dell'attività viene inclusa nel rapporto.
`TemporaryFailure`	Nell'attività si è verificato un errore temporane o e verrà reindirizzata prima del completam ento del processo. La stringa risultante viene ignorata. Se questo è l'ultimo reindirizzamento,

Codice di risposta	Descrizione
	il messaggio di errore viene incluso nel rapporto finale.
`PermanentFailure`	Nell'attività si è verificato un errore permanente. Se hai richiesto un rapporto di completamento del processo, l'attività viene contrassegnata come `Failed` e include la stringa del messaggio di errore. Le stringhe risultanti da attività non riuscite vengono ignorate.

Creazione di una funzione Lambda da utilizzare con le operazioni in batch S3

Questa sezione fornisce esempi di autorizzazioni AWS Identity and Access Management (IAM) da utilizzare con la funzione Lambda. Contiene anche una funzione Lambda di esempio da utilizzare con le operazioni in batch S3. Se non hai mai creato una funzione Lambda prima, consulta Tutorial: Using AWS Lambda with Amazon S3 nella AWS Lambda Developer Guide.

Devi creare funzioni Lambda specifiche da utilizzare con le operazioni in batch S3. Non puoi riutilizzare funzioni Lambda basate su eventi Amazon S3 esistenti. Il motivo è che le funzioni Lambda utilizzate per le operazioni in batch S3 devono accettare e restituire campi dati speciali.

> **Important**
>
> AWS Lambda le funzioni scritte in Java accettano entrambe le interfacce RequestHandlero RequestStreamHandlergestori. Tuttavia, per supportare il formato di richiesta e risposta di S3 Batch Operations, è AWS Lambda necessaria l'`RequestStreamHandler`interfaccia per la serializzazione e la deserializzazione personalizzate di una richiesta e una risposta. Questa interfaccia consente a Lambda di passare un InputStream and OutputStream al metodo Java`handleRequest`.
> Assicurati di specificare l'interfaccia `RequestStreamHandler` quando utilizzi funzioni Lambda con le operazioni in batch S3. Se utilizzi un'interfaccia `RequestHandler`, il processo batch non riuscirà restituendo il messaggio "Invalid JSON returned in Lambda payload" (JSON non valido restituito nel payload Lambda) nel report di completamento.
> Per ulteriori informazioni, consulta Interfacce Handler nella Guida per l'utente di AWS Lambda .

Autorizzazioni IAM di esempio

Di seguito sono riportati alcuni esempi delle autorizzazioni IAM necessarie per utilizzare una funzione Lambda con le operazioni in batch S3.

Example – Policy di trust delle operazioni in batch S3

Di seguito è riportato un esempio di policy di trust che puoi utilizzare per il ruolo IAM in Batch Operations. Questo ruolo IAM viene specificato quando crei il processo e concede a Batch Operations l'autorizzazione per assumere il ruolo IAM.

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

Example – Policy IAM Lambda

Di seguito è riportato un esempio di policy IAM che fornisce alle operazioni in batch S3 l'autorizzazione per richiamare la funzione Lambda e leggere il manifest di input.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BatchOperationsLambdaPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:PutObject",
                "lambda:InvokeFunction"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

Richiesta e risposta di esempio

Questa sezione fornisce esempi di richiesta e risposta per la funzione Lambda.

Example Richiesta

Di seguito è riportato un esempio JSON di richiesta per la funzione 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 Risposta

Di seguito è riportato un esempio JSON di risposta per la funzione Lambda.

```
{
  "invocationSchemaVersion": "1.0",
  "treatMissingKeysAs" : "PermanentFailure",
  "invocationId" : "YXNkbGZqYWRmaiBhc2RmdW9hZHNmZGpmaGFzbGtkaGZza2RmaAo",
  "results": [
    {
      "taskId": "dGFza2lkZ29lc2hlcmUK",
      "resultCode": "Succeeded",
      "resultString": "[\"Mary Major", \"John Stiles\"]"
    }
  ]
}
```

Funzione Lambda di esempio per le operazioni in batch S3

Nell'esempio seguente Python Lambda rimuove un contrassegno di eliminazione da un oggetto con versione.

Come mostrato nell'esempio, le chiavi di operazioni in batch S3 sono codificate in formato URL. Per utilizzare Amazon S3 con altri AWS servizi, è importante decodificare l'URL della chiave passata da S3 Batch Operations.

```
import logging
from urllib import parse
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
logger.setLevel("INFO")

s3 = boto3.client("s3")


def lambda_handler(event, context):
    """
    Removes a delete marker from the specified versioned object.

    :param event: The S3 batch event that contains the ID of the delete marker
                  to remove.
    :param context: Context about the event.
    :return: A result structure that Amazon S3 uses to interpret the result of the
             operation. When the result code is TemporaryFailure, S3 retries the
             operation.
    """
    # Parse job parameters from Amazon S3 batch operations
    invocation_id = event["invocationId"]
    invocation_schema_version = event["invocationSchemaVersion"]

    results = []
    result_code = None
    result_string = None

    task = event["tasks"][0]
    task_id = task["taskId"]

    try:
```

```
        obj_key = parse.unquote(task["s3Key"], encoding="utf-8")
        obj_version_id = task["s3VersionId"]
        bucket_name = task["s3BucketArn"].split(":")[-1]

        logger.info(
            "Got task: remove delete marker %s from object %s.", obj_version_id,
obj_key
        )

        try:
            # If this call does not raise an error, the object version is not a delete
            # marker and should not be deleted.
            response = s3.head_object(
                Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id
            )
            result_code = "PermanentFailure"
            result_string = (
                f"Object {obj_key}, ID {obj_version_id} is not " f"a delete marker."
            )

            logger.debug(response)
            logger.warning(result_string)
        except ClientError as error:
            delete_marker = error.response["ResponseMetadata"]["HTTPHeaders"].get(
                "x-amz-delete-marker", "false"
            )
            if delete_marker == "true":
                logger.info(
                    "Object %s, version %s is a delete marker.", obj_key,
obj_version_id
                )
                try:
                    s3.delete_object(
                        Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id
                    )
                    result_code = "Succeeded"
                    result_string = (
                        f"Successfully removed delete marker "
                        f"{obj_version_id} from object {obj_key}."
                    )
                    logger.info(result_string)
                except ClientError as error:
                    # Mark request timeout as a temporary failure so it will be
retried.
```

```
                if error.response["Error"]["Code"] == "RequestTimeout":
                    result_code = "TemporaryFailure"
                    result_string = (
                        f"Attempt to remove delete marker from  "
                        f"object {obj_key} timed out."
                    )
                    logger.info(result_string)
                else:
                    raise
        else:
            raise ValueError(
                f"The x-amz-delete-marker header is either not "
                f"present or is not 'true'."
            )
    except Exception as error:
        # Mark all other exceptions as permanent failures.
        result_code = "PermanentFailure"
        result_string = str(error)
        logger.exception(error)
    finally:
        results.append(
            {
                "taskId": task_id,
                "resultCode": result_code,
                "resultString": result_string,
            }
        )
    return {
        "invocationSchemaVersion": invocation_schema_version,
        "treatMissingKeysAs": "PermanentFailure",
        "invocationId": invocation_id,
        "results": results,
    }
```

Creazione di un processo di operazioni in batch Amazon S3 che richiama una funzione Lambda

Quando crei un processo di operazioni in batch Amazon S3 per richiamare una funzione Lambda, devi fornire gli elementi seguenti:

- ARN della funzione Lambda, che può includere l'alias della funzione o un numero specifico di versione
- Ruolo IAM con l'autorizzazione per richiamare la funzione
- Il parametro dell'operazione `LambdaInvokeFunction`

Per ulteriori informazioni sulla creazione di un processo di operazioni in batch Amazon S3, consulta Creazione di un processo di operazioni in batch S3 e Operazioni supportate dalle operazioni in batch S3.

L'esempio seguente crea un processo di operazioni in batch S3 che richiama una funzione Lambda tramite 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"
```

Aggiunta di informazioni a livello di attività nei manifest Lambda

Quando utilizzi AWS Lambda le funzioni con S3 Batch Operations, potresti volere che dati aggiuntivi accompagnino ogni attività/tasto su cui viene utilizzata. Ad esempio, ti potrebbe far comodo disporre di una chiave dell'oggetto di origine e di una nuova chiave dell'oggetto. La funzione Lambda può quindi copiare la chiave di origine in un nuovo bucket S3 con un nuovo nome. Per impostazione predefinita, le operazioni in batch Amazon S3 ti permettono di specificare solo il bucket di destinazione e un elenco di chiavi di origine nel manifest di input nel processo. Di seguito viene descritto come includere dati aggiuntivi nel manifest per poter eseguire funzioni Lambda più complesse.

Per specificare i parametri per chiave nel manifest delle operazioni in batch S3 da utilizzare nel codice della funzione Lambda, utilizza il formato JSON con codifica in formato URL seguente. Il

campo `key` viene passato alla funzione Lambda come se fosse una chiave oggetto Amazon S3. Tuttavia, può essere interpretato dalla funzione Lambda come contenente altri valori o più chiavi, come mostrato di seguito.

> **Note**
>
> Il numero massimo di caratteri per il campo `key` nel manifest è 1.024.

Example – Manifest in cui le chiavi Amazon S3 vengono sostituite con stringhe JSON

Alle operazioni in batch S3 deve essere fornita la versione con codifica in formato URL.

```
my-bucket,{"origKey": "object1key", "newKey": "newObject1Key"}
my-bucket,{"origKey": "object2key", "newKey": "newObject2Key"}
my-bucket,{"origKey": "object3key", "newKey": "newObject3Key"}
```

Example – Manifest con codifica in formato URL

Alle operazioni in batch S3 deve essere fornita questa versione con codifica in formato URL. La versione senza codifica URL non funziona.

```
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 – Funzione Lambda con formato manifest che scrive i risultati nel report del processo

Questo esempio di manifesto con codifica URL contiene chiavi oggetto delimitate da pipe per l'analisi della seguente funzione Lambda.

```
my-bucket,object1key%7Clower
my-bucket,object2key%7Cupper
my-bucket,object3key%7Creverse
my-bucket,object4key%7Cdelete
```

Questa funzione Lambda mostra come analizzare un'attività delimitata da pipe codificata nel manifest di operazioni in batch S3. L'attività indica quale operazione di revisione viene applicata all'oggetto specificato.

```
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,
    }
```

Informazioni sul tutorial su Operazioni in batch Amazon S3

Il seguente tutorial presenta end-to-end le procedure complete per alcune attività di Batch Operations con Lambda.

- Tutorial: transcodifica in batch di video con S3 Batch Operations e AWS LambdaAWS Elemental MediaConvert

Sostituisci tutti i tag oggetto

L'operazione Sostituisci tutti i tag oggetto sostituisce i tag oggetto di Amazon S3 su ogni oggetto riportato nel manifest. Un tag oggetto Amazon S3 è una coppia chiave-valore di stringhe che permette di archiviare i metadati di un oggetto.

Per creare un processo Sostituisci tutti i tag oggetto, devi specificare un set di tag da applicare. S3 Batch Operations applica lo stesso set di tag a ogni oggetto. Il set di tag fornito sostituisce tutti i set di tag già associati agli oggetti nel manifest. S3 Batch Operations non supporta l'aggiunta di tag agli oggetti mantenendo i tag esistenti in posizione.

Se gli oggetti nel manifest si trovano in un bucket con versione, dovrai applicare il set di tag alle versioni specifiche di ogni oggetto. A questo scopo, è necessario specificare un ID versione per ogni oggetto nel manifest. Se non includi un ID versione per ogni oggetto, allora S3 Batch Operations applicherà la lista di controllo accessi all'ultima versione dell'oggetto.

Restrizioni e limitazioni

- Il ruolo AWS Identity and Access Management (IAM) specificato per eseguire il processo di operazioni in batch deve disporre delle autorizzazioni per eseguire l'operazione Sostituisci tutti

i tag oggetto di Amazon S3. Per ulteriori informazioni sulle autorizzazioni necessarie, consulta PutObjectTagging nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

- S3 Batch Operations utilizza l'operazione PutObjectTagging di Amazon S3 per applicare tag a ogni oggetto nel manifest. Tutte le restrizioni e le limitazioni che si applicano all'operazione sottostante si applicano anche ai processi S3 Batch Operations.

Per ulteriori informazioni sull'utilizzo della console per creare processi, consulta Creazione di un processo di operazioni in batch S3.

Per ulteriori dettagli sul tag degli oggetti, consulta Suddivisione in categorie dello storage utilizzando i tag in questa guida e PutObjectTagging, GetObjectTagging e DeleteObjectTagging nel Riferimento alle API di Amazon Simple Storage Service.

Elimina tutti i tag oggetto

L'operazione Elimina tutti i tag oggetto rimuove tutti i set di tag oggetto Amazon S3 correntemente associati agli oggetti riportati nel manifest. S3 Batch Operations non supporta l'eliminazione di tag dagli oggetti mantenendo altri tag in posizione.

Se gli oggetti nel manifest si trovano in un bucket con versione, puoi rimuovere i set di tag da una versione specifica di un oggetto. A questo scopo, è necessario specificare un ID versione per ogni oggetto riportato nel manifest. Se non includi un ID versione per un oggetto, S3 Batch Operations rimuoverà il set di tag dall'ultima versione di ogni oggetto.

Per ulteriori informazioni sui manifest di Batch Operations, consulta Specifica di un manifest.

> **Warning**
>
> L'esecuzione di questo processo rimuove tutti i set di tag oggetto in ogni oggetto elencato nel manifest.

Restrizioni e limitazioni

- Il ruolo AWS Identity and Access Management (IAM) specificato per eseguire il processo deve disporre delle autorizzazioni per eseguire l'operazione Delete object tagging di Amazon S3. Per ulteriori informazioni, consulta DeleteObjectTagging nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

- S3 Batch Operations utilizza l'operazione DeleteObjectTagging di Amazon S3 per rimuovere i set di tag da ogni oggetto nel manifest. Tutte le restrizioni e le limitazioni che si applicano all'operazione sottostante si applicano anche ai processi S3 Batch Operations.

Per informazioni su come creare i processi, consulta Creazione di un processo di operazioni in batch S3.

Per ulteriori dettagli sul tagging degli oggetti, consulta Sostituisci tutti i tag oggetto in questa guida e PutObjectTagging, GetObjectTagging e DeleteObjectTagging nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Sostituisci lista di controllo degli accessi (ACL)

L'operazione Sostituisci lista di controllo degli accessi (ACL) sostituisce le liste di controllo degli accessi (ACL) di Amazon S3 per ogni oggetto riportato nel manifest. Tali liste consentono di specificare chi può accedere a un oggetto e quali operazioni può eseguire.

Le operazioni in batch S3 supportano liste di controllo accessi da te definite e liste di controllo accessi predefinite fornite da Amazon S3 con un set predefinito di autorizzazioni di accesso.

Se gli oggetti nel manifest si trovano in un bucket con versione, devi applicare le liste di controllo degli accessi alle versioni specifiche di ogni oggetto. A questo scopo, è necessario specificare un ID versione per ogni oggetto nel manifest. Se non includi un ID versione per ogni oggetto, le operazioni in batch S3 applicano la lista di controllo accessi all'ultima versione dell'oggetto.

Per ulteriori informazioni sulle ACL in Amazon S3, Panoramica delle liste di controllo accessi (ACL).

Blocco dell'accesso pubblico di S3

Se desideri limitare l'accesso pubblico a tutti gli oggetti in un bucket, utilizza il blocco dell'accesso pubblico in Amazon S3 invece di operazioni in batch S3. Il blocco dell'accesso pubblico può limitare l'accesso pubblico a livello di bucket o di account grazie a una sola semplice operazione con effetto rapido. Questa opzione è ideale se il tuo obiettivo è controllare l'accesso pubblico a tutti gli oggetti inclusi in un bucket o account. Utilizza S3 Batch Operations se devi applicare una lista ACL personalizzata a ogni oggetto nel manifest. Per ulteriori informazioni sul blocco dell'accesso pubblico di S3, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

S3 Object Ownership

Se gli oggetti nel manifesto si trovano in un bucket che utilizza l'impostazione applicata del proprietario del bucket per Object Ownership (Proprietà oggetto), l'operazione Replace access control

list (ACL) (Sostituisci lista di controllo degli accessi (ACL)) può specificare solo le ACL di oggetti che concedono il controllo completo al proprietario del bucket. L'operazione non può concedere autorizzazioni ACL dell'oggetto ad altri Account AWS o gruppi. Per ulteriori informazioni, consultare Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

Restrizioni e limitazioni

- Il ruolo specificato per eseguire il processo Sostituisci lista di controllo degli accessi deve disporre delle autorizzazioni per eseguire l'operazione `PutObjectAcl` di Amazon S3. Per ulteriori informazioni sulle autorizzazioni necessarie, consulta PutObjectAcl nella Documentazione di riferimento delle API di Amazon Simple Storage Service.
- S3 Batch Operations utilizza l'operazione `PutObjectAcl` di Amazon S3 per applicare la lista ACL specificata a ogni oggetto nel manifest. Pertanto, tutte le restrizioni e le limitazioni che si applicano all'operazione PutObjectACL sottostante si applicano anche ai processi Sostituisci lista di controllo degli accessi (ACL) di S3 Batch Operations.

Ripristino di oggetti con operazioni in batch

L'operazione Ripristino avvia le richieste di ripristino per gli oggetti Amazon S3 archiviati elencati nel manifesto. Prima di potervi accedere in tempo reale, i seguenti oggetti archiviati devono essere ripristinati:

- Oggetti archiviati nelle classi di archiviazione S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive
- Oggetti archiviati tramite la classe di storage S3 Intelligent-Tiering nei livelli di accesso di archiviazione o archiviazione profonda

L'uso di un'operazione S3 Initiate Restore Object nel processo S3 Batch Operations si traduce in una richiesta di ripristino per ogni oggetto specificato nel manifest.

> **Important**
>
> Il processo S3 Initiate Restore Object avvia solo la richiesta di ripristino degli oggetti. S3 Batch Operations riporta il processo come completo per ogni oggetto dopo l'avvio della richiesta per quell'oggetto. Amazon S3 non aggiorna il processo né ti informa in altro modo quando gli oggetti sono stati ripristinati. Tuttavia, puoi utilizzare le notifiche di eventi S3 per

ricevere notifiche quando gli oggetti sono disponibili in Amazon S3. Per ulteriori informazioni, consulta Notifiche di eventi Amazon S3.

Per creare un processo S3 Initiate Restore Object, sono disponibili i seguenti argomenti:

ExpirationInDays

Questo argomento specifica per quanto tempo l'oggetto di S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive rimane disponibile in Amazon S3. I processi Initiate Restore Object destinati agli oggetti di S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive richiedono che `ExpirationInDays` sia impostato su 1 o su un valore superiore.

> **Important**
>
> Non impostare `ExpirationInDays` durante la creazione di processi operativi S3 Initiate Restore Object destinati agli oggetti di livello Accesso archivio e Accesso archivio approfondito di S3 Intelligent-Tiering. Gli oggetti nei livelli di accesso S3 Intelligent-Tiering non sono soggetti alla scadenza del ripristino, quindi specificando `ExpirationInDays` si avrà un errore della richiesta di ripristino.

GlacierJobTier

Amazon S3 può ripristinare gli oggetti utilizzando uno dei tre diversi livelli di recupero: `EXPEDITED`, `STANDARD` e `BULK`. Tuttavia, la funzionalità S3 Batch Operations supporta solo i livelli di `STANDARD` recupero. Per ulteriori informazioni sulle differenze tra i livelli di recupero, consulta Opzioni di recupero dall'archivio.

Per ulteriori informazioni sui prezzi per ogni livello, consulta la sezione Richieste e recupero dati nella pagina dei prezzi di Amazon S3.

Differenze nel ripristino da S3 Glacier e S3 Intelligent-Tiering

Il ripristino dei file archiviati dalle classi di archiviazione S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive differisce dal ripristino dei file dalla classe di archiviazione S3 Intelligent-Tiering nei livelli Archive Access o Deep Archive Access.

- Quando esegui il ripristino da S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, viene creata una copia temporanea dell'oggetto. Amazon S3 elimina questa copia dopo che il valore specificato nell'argomento `ExpirationInDays` è trascorso. Dopo aver eliminato questa copia temporanea, dovrai inviare una richiesta di ripristino aggiuntiva per accedere all'oggetto.
- Durante il ripristino degli oggetti S3 Intelligent-Tiering archiviati.non specificare l'argomento `ExpirationInDays`. Quando esegui il ripristino di un oggetto dai livelli di accesso Accesso archivio e Accesso archivio approfondito di S3 Intelligent-Tiering, l'oggetto passa nuovamente al livello Frequent Access di S3 Intelligent-Tiering. Dopo un minimo di 90 giorni consecutivi senza accesso, l'oggetto passa automaticamente al livello Accesso archivio. Dopo un minimo di 180 giorni consecutivi senza accesso, l'oggetto passa automaticamente al livello Accesso archivio approfondito.
- I processi delle operazioni in batch possono funzionare su oggetti di classe di archiviazione S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive o su oggetti di livello di archiviazione Accesso archivio e Accesso archivio approfondito di S3 Intelligent-Tiering. Le operazioni in batch non possono operare su entrambi i tipi di oggetti archiviati nello stesso processo. Per ripristinare oggetti di entrambi i tipi, devi creare processi Batch Operations separati.

Ripristini sovrapposti

Se il processo S3 Initiate Restore Object prova a ripristinare un oggetto già in fase di ripristino, S3 Batch Operations si comporta nel seguente modo.

L'operazione di ripristino dell'oggetto riesce se una di queste condizioni restituisce true:

- Confrontato alla richiesta di ripristino già in corso, il valore `ExpirationInDays` del processo è uguale e il relativo valore `GlacierJobTier` è più veloce.
- La richiesta di ripristino precedente è già stata completata e l'oggetto è attualmente disponibile. In questo caso, le operazioni in batch aggiornano la data di scadenza dell'oggetto ripristinato in modo che corrisponda al valore `ExpirationInDays` specificato nella richiesta di ripristino in corso.

L'operazione di ripristino dell'oggetto non riesce se una o più delle seguenti condizioni restituisce true:

- La richiesta di ripristino già in corso non è stata ancora completata e la durata del ripristino per questo processo, specificata dal valore `ExpirationInDays`, è diversa dalla durata specificata nella richiesta di ripristino in corso.
- Il livello di ripristino per questo processo (specificato dal valore `GlacierJobTier`),è uguale o inferiore al livello di ripristino specificato nella richiesta di ripristino in corso.

Limitazioni

I processi S3 Initiate Restore Object presentano le seguenti limitazioni:

- Devi creare il processo nella stessa regione degli oggetti archiviati.
- S3 Batch Operations non supporta il livello di recupero `EXPEDITED`.

Per ulteriori informazioni sul ripristino degli oggetti, consulta Ripristino di un oggetto archiviato.

Conservazione Blocco oggetto S3

L'operazione Conservazione Blocco oggetti consente di applicare date di conservazione per gli oggetti utilizzando la modalità di governance o la modalità di conformità. Queste modalità di conservazione applicano livelli di protezione diversi. È possibile applicare entrambe le modalità di conservazione a qualsiasi versione di oggetto. Le date di conservazione, ad esempio per un blocco di carattere legale, impediscono di sovrascrivere o eliminare un oggetto. Amazon S3 archivia la data di fine della conservazione specificata nei metadati dell'oggetto e protegge la versione specificata dell'oggetto fino alla scadenza del periodo di conservazione.

Puoi utilizzare le operazioni in batch S3 con il blocco oggetti per gestire le date di conservazione di molti oggetti Amazon S3 contemporaneamente. Specifica l'elenco degli oggetti di destinazione nel manifest e inviarlo alle operazioni in batch per il completamento. Per ulteriori informazioni, consulta del blocco oggetti S the section called “Periodi di conservazione”.

Il processo di operazioni in batch Amazon S3 con date di conservazione viene eseguito fino al completamento, all'annullamento o al raggiungimento di uno stato di errore. È consigliabile utilizzare la conservazione con le operazioni in batch S3 e il blocco oggetti S3 per aggiungere, modificare o rimuovere la data di conservazione per molti oggetti con una singola richiesta.

Le operazioni in batch verificano che sia abilitato il blocco oggetti nel bucket prima di elaborare qualsiasi chiave nel manifest. Per eseguire le operazioni e la convalida, le operazioni in batch devono avere le autorizzazioni `s3:GetBucketObjectLockConfiguration` e `s3:PutObjectRetention` in un ruolo IAM per poter chiamare il blocco oggetti per conto tuo. Per ulteriori informazioni, consulta the section called “Considerazioni su Object Lock”.

Per informazioni sull'utilizzo di questa operazione con l'API REST, consulta `S3PutObjectRetention` nell'operazione CreateJob nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Per un esempio AWS Command Line Interface di utilizzo di questa operazione, consulta the section called “Utilizzare le operazioni in batch con la conservazione il blocco oggetti”. Per un esempio, AWS SDK for Java consulta the section called “Utilizzare le operazioni in batch con la conservazione il blocco oggetti”.

Restrizioni e limitazioni

- Le operazioni in batch S3 non apportano alcuna modifica a livello di bucket.
- La funzione Versioni multiple e il blocco oggetti S3 devono essere configurati nel bucket in cui viene eseguito il processo.
- Tutti gli oggetti elencati nel manifest devono trovarsi nello stesso bucket.
- L'operazione funziona sulla versione più recente dell'oggetto, a meno che non venga specificata esplicitamente una versione nel manifest.
- Per utilizzarla, devi avere l'autorizzazione `s3:PutObjectRetention` nel ruolo IAM.
- `s3:GetBucketObjectLockConfiguration`L'autorizzazione IAM è necessaria per verificare che il blocco oggetti sia abilitato per il bucket S3.
- È possibile estendere solo il periodo di conservazione di oggetti con applicate date di conservazione in modalità `COMPLIANCE` e non può essere abbreviato.

Blocco di carattere legale del blocco oggetti S3

L'operazione Blocco di carattere legale del blocco oggetti consente di porre un blocco di carattere legale alla versione di un oggetto. Analogamente all'impostazione di un periodo di conservazione, un blocco a di carattere legale impedisce che una versione di un oggetto venga sovrascritta o eliminata. Tuttavia, un blocco a fini giudiziari non ha un periodo di conservazione associato e rimane valido fino a quando non viene rimosso.

Puoi utilizzare S3 Batch Operations con il blocco oggetti per aggiungere blocchi di carattere legale a molti oggetti Amazon S3 contemporaneamente. A questo scopo, puoi elencare gli oggetti di destinazione nel manifest e inviare l'elenco a Batch Operations. Il processo di operazioni in batch Amazon S3 con il blocco di carattere legale del blocco oggetti viene eseguito fino al completamento, all'annullamento o al raggiungimento di uno stato di errore.

Le operazioni in batch S3 verificano che il blocco oggetti sia abilitato nel bucket S3 prima di elaborare qualsiasi chiave nel manifest. Per eseguire le operazioni sugli oggetti e la convalida a livello di bucket, S3 Batch Operations necessita di `s3:PutObjectLegalHold` e

`s3:GetBucketObjectLockConfiguration` in un ruolo IAM in modo da poter richiamare il blocco di oggetti S3 per tuo conto.

Quando crei il progetto di operazioni in batch S3 per rimuovere il blocco di carattere legale, devi solo specificare Off (Disattivato) come stato del blocco. Per ulteriori informazioni, consulta the section called “Considerazioni su Object Lock”.

Per informazioni su come utilizzare questa operazione con l'API REST, consulta `S3PutObjectLegalHold` nell'operazione CreateJob nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Per un esempio di utilizzo di questa operazione, consulta Utilizzo dell' AWS SDK for Java.

Restrizioni e limitazioni

- Le operazioni in batch S3 non apportano alcuna modifica a livello di bucket.
- Tutti gli oggetti elencati nel manifest devono trovarsi nello stesso bucket.
- La funzione Versioni multiple e il blocco oggetti S3 devono essere configurati nel bucket in cui viene eseguito il processo.
- L'operazione funziona sulla versione più recente dell'oggetto, a meno che non venga specificata esplicitamente una versione nel manifest.
- `s3:PutObjectLegalHold`Per aggiungere o rimuovere un blocco di carattere legale negli oggetti, è necessaria l'autorizzazione nel ruolo IAM.
- `s3:GetBucketObjectLockConfiguration`L'autorizzazione IAM è necessaria per verificare che il blocco oggetti S3 sia abilitato per il bucket S3.

- Copia oggetti
- Funzione Invoke AWS Lambda
- Sostituisci tutti i tag oggetto
- Elimina tutti i tag oggetto
- Sostituisci lista di controllo degli accessi (ACL)
- Ripristino di oggetti con operazioni in batch
- Conservazione Blocco oggetto S3
- Blocco di carattere legale del blocco oggetti S3
- Replica di oggetti esistenti con S3 Batch Replication

Gestione dei processi di operazioni in batch Amazon S3

Amazon S3 fornisce un valido set di strumenti che consentono di gestire i processi di operazioni in batch S3 dopo la loro creazione. In questa sezione vengono descritte le operazioni che è possibile utilizzare per gestire e tenere traccia dei processi utilizzando la AWS Management Console, la AWS CLI, gli SDK AWS o l'API REST.

Argomenti

Utilizzo della console Amazon S3 per gestire i processi S3 Batch Operations

Puoi gestire i processi S3 Batch Operations utilizzando la console. Ad esempio, puoi:

- Visualizzare i processi attivi e in coda
- Modificare la priorità di un processo
- Confermare ed eseguire un processo
- Clonazione di un processo
- Annullamento di un processo

Gestione di Batch Operations tramite la console

1. Accedi alla AWS Management Console e apri la console di Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Operazioni in batch.
3. Scegli il processo specifico che desideri gestire.

Elenchi di processi

Puoi recuperare un elenco dei tuoi processi di operazioni in batch Amazon S3. L'elenco include processi non ancora completati, nonché processi completati negli ultimi 90 giorni. L'elenco di processi include informazioni per ogni processo, quali ID, descrizione, priorità, stato corrente e numero di

attività riuscite e non riuscite. Puoi filtrare l'elenco dei processi in base allo stato. Quando recuperi un elenco di processi tramite la console, puoi anche cercare i processi in base alla descrizione o all'ID e filtrarli in base alla Regione AWS.

Ottenimento di un elenco di processi attivi e completi

Nell'esempio AWS CLI seguente viene visualizzato un elenco di processi `Active` e `Complete`.

```
aws s3control list-jobs \
    --region us-west-2 \
    --account-id acct-id \
    --job-statuses '["Active","Complete"]' \
    --max-results 20
```

Per ulteriori informazioni ed esempi, consulta list-jobs nel Riferimento ai comandi AWS CLI.

Visualizzazione dettagli processo

Per ulteriori informazioni su un processo che puoi recuperare elencando i processi, visualizza tutti i dettagli per un singolo processo. Puoi visualizzare i dettagli per i processi non ancora completati o i processi completati negli ultimi 90 giorni. Oltre alle informazioni restituite in un elenco di processi, i dettagli di un singolo processo includono altri elementi come:

- I parametri operativi
- Dettagli sul manifesto
- Informazioni sul report di completamento (se ne hai configurato uno al momento della creazione del processo)
- L'Amazon Resource Name (ARN) del ruolo utente a cui hai assegnato l'esecuzione del processo

Visualizzando i dettagli di un singolo processo, puoi accedere all'intera configurazione del processo. Per visualizzare i dettagli di un processo, puoi utilizzare la console Amazon S3 o la AWS Command Line Interface (AWS CLI).

Ottenere la descrizione di un processo Operazioni in batch Amazon S3 nella console Amazon S3

Per visualizzare una descrizione del processo Operazioni in batch utilizzando la console

1. Accedi alla AWS Management Console e apri la console di Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

2. Nel pannello di navigazione a sinistra, scegli Operazioni in batch.
3. Scegli l'ID del processo specifico per visualizzarne i dettagli.

Ottenere la descrizione di un processo Operazioni in batch Amazon S3 nella AWS CLI

Nell'esempio seguente viene recuperata la descrizione di un processo Operazioni in batch Amazon S3 tramite la AWS CLI. Per utilizzare il seguente comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
aws s3control describe-job \
--region us-west-2 \
--account-id acct-id \
--job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

Per ulteriori informazioni ed esempi, consulta describe-job nel Riferimento ai comandi AWS CLI.

Assegnazione della priorità dei processi

È possibile assegnare a ciascun processo una priorità numerica, che può essere qualsiasi numero intero positivo. Le operazioni in batch S3 danno la precedenza ai processi in base alla priorità assegnata. I processi con priorità superiore (o un valore numero più alto per il parametro di priorità) vengono valutati per primi. La priorità viene determinata in ordine decrescente, ad esempio a una coda di processi con valore di priorità 10 viene assegnata una preferenza di pianificazione rispetto a una coda di processi con valore di priorità 1.

Puoi modificare la priorità di un processo mentre è in esecuzione. Se invii un nuovo processo con una priorità più alta mentre un processo è in esecuzione, il processo di priorità inferiore può essere sospeso per consentire l'esecuzione del processo con priorità più alta.

La modifica della priorità del lavoro non influisce sulla velocità di elaborazione dei processi.

> **Note**
>
> Le operazioni in batch S3 rispettano le priorità dei processi sulla base del miglior tentativo. Sebbene i processi con priorità più alta abbiano la precedenza sui processi con priorità più bassa, Amazon S3 non garantisce un ordinamento rigoroso dei processi.

Utilizzo della console S3

Come aggiornare la priorità del processo nella AWS Management Console

1. Accedi alla AWS Management Console e apri la console di Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Operazioni in batch.
3. Scegli il processo specifico che desideri gestire.
4. Scegli Actions (Operazioni). Nell'elenco a discesa, scegli Update priority (Aggiorna priorità).

Utilizzo di AWS CLI

Nell'esempio seguente viene aggiornata la priorità dei processi utilizzando l AWS CLI. Un numero più alto indica una priorità di esecuzione più alta.

```
aws s3control update-job-priority \
    --region us-west-2 \
    --account-id acct-id \
    --priority 98 \
    --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

Utilizzo di AWS SDK for Java

Nell'esempio seguente viene aggiornata la priorità di un processo di operazioni in batch S3 tramite la AWS SDK for Java.

Per ulteriori informazioni sulla priorità dei processi, consulta Assegnazione della priorità dei processi.

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();
        }
    }
}
```

Monitoraggio dei rapporti sullo stato e sul completamento dei processi

Con le operazioni in batch S3 è possibile visualizzare e aggiornare lo stato del processo, aggiungere notifiche e registrazione, tenere traccia degli errori di processo e generare report di completamento.

Argomenti

- Stati del processo
- Aggiornamento dello stato del processo
- Notifiche e registrazione
- Monitoraggio degli errori dei processi

- Rapporti di completamento
- Esempi: Monitoraggio di un processo di operazioni in batch S3 in Amazon EventBridge tramite AWS CloudTrail
- Esempi: report di completamento delle operazioni in batch S3

Stati del processo

Una volta creato e avviato, un processo passa attraverso una serie di stati. Nella tabella seguente vengono descritti gli stati e le possibili transizioni tra di essi.

Stato	Descrizione	Transizioni
`New`	Quando viene creato, un processo è nello stato `New`.	Un processo passa automaticamente allo stato `Preparing` quando Amazon S3 inizia a elaborare l'oggetto manifest.
`Preparing`	Amazon S3 sta elaborando l'oggetto manifest e altri parametri allo scopo di configurare ed eseguire il processo.	Un processo passa automaticamente allo stato `Ready` quando Amazon S3 termina l'elaborazione del manifest e di altri parametri. A questo punto è pronto per iniziare a eseguire l'operazione specificata sugli oggetti elencati nel file manifest. Se il processo richiede una conferma prima dell'esecuzione, ad esempio quando si crea un processo mediante la console Amazon S3, il processo passa dallo stato `Preparing` allo stato `Suspended`. Rimane nello

Stato	Descrizione	Transizioni
		stato `Suspended` finché non confermi di eseguirlo.
`Suspended`	Il processo richiede una conferma, ma l'utente non ha ancora confermato di eseguirlo. La conferma viene richiesta solo per i processi creati mediante la console Amazon S3. Un processo creato mediante la console entra nello stato `Suspended` subito dopo lo stato `Preparing` . Dopo aver confermato l'esecuzione del processo e quando il processo passa allo stato `Ready`, non torna mai allo stato `Suspended` .	Una volta confermata l'esecuzione del processo, lo stato cambia in `Ready`.
`Ready`	Amazon S3 è pronto per iniziare l'esecuzione delle operazioni richieste sugli oggetti.	Un processo passa automaticamente allo stato `Active` quando Amazon S3 inizia a eseguirlo. La quantità di tempo durante il quale un processo rimane nello stato `Ready` dipende dalla presenza o meno di processi con priorità più alta già in esecuzione e dal tempo necessario per completare questi processi.

Stato	Descrizione	Transizioni
`Active`	Amazon S3 sta eseguendo un'operazione richiesta sugli oggetti elencati nel manifest. Mentre un lavoro è in corso`Active`, puoi monitorarne l'avanzamento utilizzando la console Amazon S3 o il `DescribeJob` funzionamento tramite l'API REST o AWS gli AWS CLI SDK.	Un processo esce dallo stato `Active` quando non esegue più operazioni sugli oggetti. Questo può avvenire automaticamente, ad esempio in seguito all'esito positivo o negativo di un processo, oppure può essere il risultato di azioni dell'utente come l'annullamento di un processo. Lo stato successivo del processo dipende dal motivo della transizione.
`Pausing`	Il processo sta passando allo stato `Paused` da un altro stato.	Un processo passa automatic amente allo stato `Paused` al termine della fase `Pausing`.
`Paused`	Un processo può passare allo stato `Paused` se si invia un altro processo con priorità più alta mentre il processo corrente è in esecuzione.	Un processo `Paused` torna automaticamente allo stato `Active` quando i processi con priorità più alta che ne stanno bloccando l'esecuzione vengono completati, sospesi o hanno esito negativo.

Stato	Descrizione	Transizioni
`Complete`	Il processo ha terminato l'esecuzione dell'operazione richiesta su tutti gli oggetti elencati nel manifest. Per ogni oggetto, l'operazione potrebbe aver avuto esito positivo o negativo. Se il processo è stato configurato in modo da generare un rapporto di completamento, tale rapporto sarà disponibile non appena il processo sarà passato allo stato `Complete`.	`Complete` è uno stato terminale. Quando un processo raggiunge lo stato `Complete`, non esegue più la transizione ad altri stati.
`Cancelling`	Il processo sta passando allo stato `Cancelled` .	Un processo passa automatic amente allo stato `Cancelled` al termine della fase `Cancelling` .
`Cancelled`	Hai richiesto l'annullamento del processo e le operazioni in batch S3 lo hanno eseguito con successo. Il processo non invierà nuove richieste ad Amazon S3.	`Cancelled` è uno stato terminale. Quando un processo raggiunge lo stato `Cancelled` , non esegue più la transizione ad altri stati.
`Failing`	Il processo sta passando allo stato `Failed`.	Un processo passa automatic amente allo stato `Failed` al termine della fase `Failing`.

Stato	Descrizione	Transizioni
`Failed`	Il processo ha avuto esito negativo e non è più in esecuzione. Per ulteriori informazioni sugli errori dei processi, consulta Monitoraggio degli errori dei processi.	`Failed` è uno stato terminale. Quando un processo raggiunge lo stato `Failed`, non esegue più la transizione ad altri stati.

Aggiornamento dello stato del processo

Gli esempi seguenti AWS CLI e quelli di SDK for Java aggiornano lo stato di un processo Batch Operations. Per ulteriori informazioni sull'utilizzo della console S3 per gestire i processi delle operazioni in batch, consulta Utilizzo della console Amazon S3 per gestire i processi S3 Batch Operations.

Utilizzando il AWS CLI

- Se non è stato specificato il parametro `--no-confirmation-required` nel precedente esempio `create-job`, il processo rimane in uno stato di sospensione finché non viene confermato impostando lo stato su `Ready`. Amazon S3 rende quindi il processo idoneo per l'esecuzione.

```
aws s3control update-job-status \
    --region us-west-2 \
    --account-id 181572960644 \
    --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c \
    --requested-job-status 'Ready'
```

- Annullare il processo impostandone lo stato su `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
```

Utilizzo dell' AWS SDK for Java

Nell'esempio seguente viene aggiornato lo stato di un processo di operazioni in batch S3 tramite la AWS SDK for Java.

Per ulteriori informazioni sullo stato dei processi, consulta Monitoraggio dei rapporti sullo stato e sul completamento dei processi.

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();
        }
    }
}
```

Notifiche e registrazione

Oltre a richiedere report di completamento, è anche possibile acquisire, esaminare e controllare l'attività delle operazioni in batch utilizzando AWS CloudTrail. Poiché Batch Operations utilizza le API Amazon S3 esistenti per eseguire le attività, tali attività generano gli stessi eventi di quando vengono chiamate direttamente. Pertanto, puoi tracciare e registrare l'avanzamento del processo e di tutte le attività relative utilizzando gli stessi strumenti e processi di notifica, registrazione e controllo già utilizzati con Amazon S3. Per ulteriori informazioni, consulta gli esempi nelle sezioni seguenti.

> **Note**
>
> Amazon S3 Batch Operations genera eventi di gestione e dati CloudTrail durante l'esecuzione del processo. Il volume di questi eventi viene ridimensionato con il numero di chiavi nel manifest di ogni processo. Per ulteriori informazioni, consulta la pagina CloudTrail dei prezzi, che include esempi di come i prezzi variano in base al numero di percorsi configurati nel tuo account. Per informazioni su come configurare e registrare gli eventi in base alle tue esigenze, consulta Creazione del primo trail nella Guida per l'utente di AWS CloudTrail .

Per ulteriori informazioni sugli eventi Amazon S3, consulta Notifiche di eventi Amazon S3.

Monitoraggio degli errori dei processi

Se si verifica un problema con un processo di operazioni in batch Amazon S3 che ne impedisce l'esecuzione, ad esempio non riesce a leggere il manifest specificato, il processo non riesce. Quando un processo non riesce, genera uno o più codici o motivi di errore. Le operazioni in batch S3 archiviano i codici e i motivi di errore con il processo in modo da poterli visualizzare richiedendo i dettagli del processo. Se è stato richiesto un rapporto di completamento per il processo, deve contenere anche i codici e i motivi di errore.

Per impedire che i processi eseguano un numero elevato di operazioni non riuscite, Amazon S3 impone una soglia di errore attività su ogni processo di operazioni in batch S3. Dopo che un processo

ha eseguito almeno 1000 attività, Amazon S3 monitora il tasso di errore delle attività. Se, in qualsiasi momento, la percentuale di errore (il numero di attività non andate a buon fine espresso come proporzione del numero totale di attività eseguite) supera il 50%, il lavoro non riesce. Se il processo non riesce perché ha superato la soglia di errore attività, è possibile identificare la causa degli errori. È ad esempio possibile che nel manifest siano stati involontariamente inclusi alcuni oggetti che non esistono nel bucket specificato. Dopo aver risolto gli errori, è possibile inviare nuovamente il processo.

> **Note**
>
> Le operazioni in batch S3 funzionano in modo asincrono e non eseguono necessariamente le attività nell'ordine in cui gli oggetti sono elencati nel manifest. Pertanto non puoi utilizzare l'ordinamento del manifest per determinare quali attività degli oggetti sono riuscite e quali no. Puoi invece esaminare il rapporto di completamento del lavoro (se ne hai richiesto uno) o visualizzare i registri degli AWS CloudTrail eventi per determinare l'origine degli errori.

Rapporti di completamento

Quando crei un processo, puoi richiedere un rapporto di completamento. Fintantoché le operazioni in batch S3 invocano correttamente almeno un'attività, Amazon S3 genera un report di completamento quando termina l'esecuzione dell'attività o quando non riesce a eseguirla o viene annullato. Puoi configurare il rapporto di completamento per includere tutte le attività o solo quelle non riuscite.

Il rapporto di completamento include la configurazione del processo e lo stato e le informazioni per ogni attività, inclusi la chiave e la versione dell'oggetto, lo stato, i codici di errore e le descrizioni degli errori. I report di completamento offrono un modo semplice per visualizzare i risultati delle attività in un formato consolidato, senza ulteriori operazioni di configurazione. I report di completamento sono crittografati con chiavi gestite di Amazon S3 (SSE-S3). Per un esempio di report di completamento, consulta Esempi: report di completamento delle operazioni in batch S3.

Se non configuri un rapporto di completamento, puoi comunque monitorare e controllare il tuo lavoro e le relative attività utilizzando CloudTrail e Amazon CloudWatch. Per ulteriori informazioni, consulta la sezione seguente.

Argomenti

- Esempi: Monitoraggio di un processo di operazioni in batch S3 in Amazon EventBridge tramite AWS CloudTrail

- Esempi: report di completamento delle operazioni in batch S3

Esempi: Monitoraggio di un processo di operazioni in batch S3 in Amazon EventBridge tramite AWS CloudTrail

L'attività dei processi di operazioni in batch Amazon S3 viene registrata in forma di eventi in AWS CloudTrail. È possibile creare una regola personalizzata in Amazon EventBridge e inviare questi eventi alla risorsa di notifica di destinazione scelta, ad esempio Amazon Simple Notification Service (Amazon SNS).

> **Note**
>
> Amazon EventBridge è il metodo preferito per gestire gli eventi. Amazon CloudWatch Events ed EventBridge sono lo stesso servizio e la stessa API di base, ma EventBridge fornisce più funzionalità. Le modifiche apportate in CloudWatch o EventBridge verranno visualizzate in ciascuna console. Per ulteriori informazioni, consulta la Guida per l'utente di Amazon EventBridge.

Esempi di monitoraggio

- Eventi delle operazioni in batch S3 registrati in CloudTrail
- Utilizzo di una regola EventBridge per il monitoraggio degli eventi di processo di operazioni in batch S3

Eventi delle operazioni in batch S3 registrati in CloudTrail

Quando viene creato un processo di operazioni in batch, viene registrato come evento `JobCreated` in CloudTrail. Durante l'esecuzione del processo, lo stato cambia durante l'elaborazione e in CloudTrail vengono registrati altri eventi `JobStatusChanged`. È possibile visualizzare questi eventi sulla console di CloudTrail. Per ulteriori informazioni su CloudTrail, consulta la Guida per l'utente di AWS CloudTrail.

> **Note**
>
> Solo gli eventi `status-change` di processo di operazioni in batch S3 vengono registrati in CloudTrail.

Example Evento di completamento del processo di operazioni in batch S3 registrato da 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": []
    }
}
```

Utilizzo di una regola EventBridge per il monitoraggio degli eventi di processo di operazioni in batch S3

Nell'esempio seguente viene illustrato come creare una regola in Amazon EventBridge per acquisire gli eventi delle operazioni in batch S3 registrati da AWS CloudTrail su una destinazione desiderata.

A tale scopo, crea una regola seguendo tutti i passaggi descritti in Creazione di regole EventBridge che reagiscono agli eventi. È possibile incollare la seguente policy personalizzata di modello di eventi delle operazioni in batch S3, se applicabile, e scegliere il servizio di destinazione desiderato.

Policy personalizzata di modello di eventi delle operazioni in batch S3

```
{
    "source": [
        "aws.s3"
    ],
    "detail-type": [
        "AWS Service Event via CloudTrail"
    ],
    "detail": {
        "eventSource": [
            "s3.amazonaws.com"
        ],
        "eventName": [
            "JobCreated",
            "JobStatusChanged"
        ]
    }
}
```

Gli esempi seguenti sono due eventi di operazioni in batch inviati ad Amazon Simple Queue Service (Amazon SQS) da una regola evento EventBridge. Un processo di operazioni in batch attraversa molti stati diversi durante l'elaborazione (`New`, `Preparing`, `Active` e così via), quindi è possibile ricevere diversi messaggi per ogni processo.

Example Evento campione di 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 di completamento del processo 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": []
    }
  }
}
```

Esempi: report di completamento delle operazioni in batch S3

Quando crei un processo di operazioni in batch S3, è possibile richiedere un report di completamento per tutte le attività o solo per le attività non andate a buon fine. Se almeno un'attività è stata invocata correttamente, le operazioni in batch S3 generano un report per i processi che sono stati completati, che non sono andati a buon fine o che sono stati annullati.

Il rapporto di completamento contiene informazioni aggiuntive per ogni attività, inclusi il nome della chiave e la versione dell'oggetto, lo stato, i codici di errore e le descrizioni degli errori. La descrizione degli errori per ogni attività non andata a buon fine può essere utilizzata per diagnosticare problemi durante la creazione del lavoro, come le autorizzazioni.

> **Note**
>
> I report di completamento sono sempre crittografati con chiavi gestite di Amazon S3 (SSE-S3).

Example file dei risultati manifest di primo livello

Il file `manifest.json` di primo livello contiene le posizioni di tutti i rapporti andati a buon fine e (se il processo conteneva errori) la posizione dei rapporti non andati a buon fine, come mostrato nel seguente esempio.

```
{
    "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 report sulle attività non andate a buon fine

I rapporti sulle attività non riuscite contengono le seguenti informazioni per tutte le attività non andate a buon fine:

- `Bucket`
- `Key`
- `VersionId`
- `TaskStatus`
- `ErrorCode`
- `HTTPStatusCode`
- `ResultMessage`

Il seguente report di esempio mostra un caso in cui la AWS Lambda funzione è scaduta, causando errori che superano la soglia di errore. È stato quindi contrassegnato come `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 report sulle attività andate a buon fine

I report sulle attività completate contengono quanto segue per le attività riuscite:

- `Bucket`
- `Key`
- `VersionId`
- `TaskStatus`
- `ErrorCode`
- `HTTPStatusCode`
- `ResultMessage`

Nel seguente esempio, la funzione Lambda ha copiato correttamente l'oggetto Amazon S3 in un altro bucket. La risposta di Amazon S3 restituita viene passata alle operazioni in batch S3 e quindi viene scritta nel report di completamento finale.

```
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'}}}"
```

Controllo dei lavori di accesso ed etichettatura mediante tag

È possibile etichettare e controllare l'accesso ai processi di operazioni in batch Amazon S3 aggiungendo tag. I tag possono essere utilizzati per identificare chi è responsabile di un processo di operazioni in batch. La presenza dei tag dei lavori può consentire o limitare la capacità di un utente di cancellare un lavoro, attivare un lavoro in stato di conferma o cambiare il livello di priorità di un lavoro. È possibile creare lavori con tag ad essi associati e aggiungere tag ai lavori dopo la creazione. Ogni tag è una coppia chiave-valore che può essere inclusa quando si crea il lavoro o si aggiorna in un secondo momento.

> **Warning**
>
> I tag di lavoro non devono contenere informazioni riservate o dati personali.

Considera il seguente esempio di tag: supponiamo che desideri che il reparto Finanze crei un processo Batch Operations. È possibile scrivere una policy AWS Identity and Access Management (IAM) che consente a un utente di richiamare `CreateJob`, a condizione che il lavoro venga creato con il tag `Department` assegnato al valore `Finance`. Inoltre, è possibile allegare tale policy a tutti gli utenti membri del dipartimento Finanze.

Continuando con questo esempio, è possibile scrivere una policy che consenta a un utente di aggiornare la priorità di qualsiasi lavoro con i tag desiderati o annullare qualsiasi lavoro con tali tag. Per ulteriori informazioni, consulta the section called “Controllo delle autorizzazioni”.

È possibile aggiungere tag a nuovi processi di operazioni in batch Amazon S3 al momento della loro creazione o a processi esistenti.

Sono valide le seguenti limitazioni sui tag:

- È possibile associare fino a 50 tag a un lavoro purché abbiano chiavi tag univoche.
- Una chiave di tag può essere composta da un massimo di 128 caratteri Unicode e i valori di tag possono essere composti da un massimo di 256 caratteri Unicode.
- La chiave e i valori fanno distinzione tra maiuscole e minuscole.

Per ulteriori informazioni sui limiti dei tag, consulta Restrizioni sui tag definiti dall'utente nella Guida per l'utente AWS Billing and Cost Management.

Operazioni API correlate al tagging dei processi di operazioni in batch Amazon S3

Amazon S3 supporta le seguenti operazioni API specifiche del tagging dei processi di operazioni in batch Amazon S3:

- GetJobTagging: restituisce il set di tag associato a un processo Batch Operations.
- PutJobtagging: sostituisce il set di tag associato a un processo. La gestione di tag dei processi di operazioni in batch Amazon S3 mediante questa operazione API prevede due scenari distinti:
  - Il processo non ha tag: è possibile aggiungere un set di tag a un processo (il processo non ha tag precedenti).
  - Il processo ha un set di tag esistente: per modificare il set di tag esistente, è possibile sostituirlo completamente oppure recuperarlo mediante GetJobTagging apportare modifiche al suo interno e utilizzare questa operazione API per sostituire il set di tag con quello modificato.

> **Note**
>
> Se invii questa richiesta con un set di tag vuoto, le operazioni in batch S3 eliminano il set di tag esistenti sull'oggetto. Se si utilizza questo metodo, verrà addebitata una richiesta Tier 1 (PUT). Per ulteriori informazioni, consulta Prezzi di Amazon S3.
> Per eliminare i tag esistenti per il processo Batch Operations, l'operazione `DeleteJobTagging` è da preferire perché ottiene lo stesso risultato senza incorrere in addebiti.

- DeleteJobTagging: elimina il set di tag associato a un processo Batch Operations.

Creazione di un processo Batch Operations con tag di processo utilizzati per l'etichettatura

È possibile etichettare e controllare l'accesso ai processi di operazioni in batch Amazon S3 aggiungendo tag. I tag possono essere utilizzati per identificare chi è responsabile di un processo di operazioni in batch. È possibile creare lavori con tag ad essi associati e aggiungere tag ai lavori dopo la creazione. Per ulteriori informazioni, consulta la sezione the section called “Utilizzo dei tag”.

Utilizzo di AWS CLI

Nell'esempio seguente di AWS CLI viene creato un processo `S3PutObjectCopy` di operazioni in batch S3 utilizzando i tag di processo come etichette per il processo.

1. Selezionare l'operazione o `OPERATION` da far eseguire al processo di operazioni in batch e scegliere il proprio `TargetResource`.

```
read -d '' OPERATION <<EOF
{
  "S3PutObjectCopy": {
    "TargetResource": "arn:aws:s3:::destination-bucket"
  }
}
EOF
```

2. Identificare il lavoro `TAGS` che si desidera per il lavoro. In questo caso, si applicano due tag `department` e `FiscalYear`, con i valori `Marketing` e `2020` rispettivamente.

```
read -d '' TAGS <<EOF
```

```
[
  {
    "Key": "department",
    "Value": "Marketing"
  },
  {
    "Key": "FiscalYear",
    "Value": "2020"
  }
]
EOF
```

3. Specificare `MANIFEST` per il processo di operazioni in batch.

```
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. Configurare `REPORT` per il processo di operazioni in batch.

```
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. Eseguire l'operazione `create-job` per creare il processo di operazioni in batch con input impostati nelle fasi precedenti.

```
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";
```

Utilizzo dell'SDK AWS per Java

Example

Nell'esempio seguente viene creato un processo di operazioni in batch S3 con tag tramite la 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();
}
```

Eliminazione dei tag da un processo S3 Batch Operations

È possibile utilizzare questi esempi per eliminare i tag da un processo di operazioni in batch.

Utilizzo di AWS CLI

Nell'esempio seguente vengono eliminati i tag da un processo di operazioni in batch tramite la AWS CLI.

```
aws \
    s3control delete-job-tagging \
    --account-id 123456789012 \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
    --region us-east-1;
```

Eliminare i tag di processo di un processo di operazioni in batch

Example

Nell'esempio seguente vengono eliminati i tag di un processo di operazioni in batch S3 tramite la 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);
}
```

Aggiunta dei tag di processo a un processo S3 Batch Operations esistente

È possibile utilizzare PutJobTagging per aggiungere tag di processo ai processi di operazioni in batch S3 esistenti. Per maggiori informazioni, consulta i seguenti esempi.

Utilizzo di AWS CLI

Di seguito è riportato un esempio di utilizzo di `s3control put-job-tagging` per aggiungere tag di processo al processo di operazioni in batch S3 tramite la AWS CLI.

> **Note**
>
> Se invii questa richiesta con un set di tag vuoto, le operazioni in batch S3 eliminano il set di tag esistenti sull'oggetto. Inoltre, se si utilizza questo metodo, verrà addebitata una richiesta Tier 1 (PUT). Per ulteriori informazioni, consulta la sezione Prezzi di Amazon S3.
> Per eliminare i tag esistenti per il processo Batch Operations, l'operazione `DeleteJobTagging` è da preferire perché ottiene lo stesso risultato senza incorrere in addebiti.

1. Identificare il lavoro `TAGS` che si desidera per il lavoro. In questo caso, si applicano due tag `department` e `FiscalYear`, con i valori `Marketing` e `2020` rispettivamente.

```
read -d '' TAGS <<EOF
[
  {
    "Key": "department",
    "Value": "Marketing"
  },
  {
    "Key": "FiscalYear",
    "Value": "2020"
  }
]
EOF
```

2. Eseguire l'operazione `put-job-tagging` con i parametri richiesti.

```
aws \
    s3control put-job-tagging \
    --account-id 123456789012 \
    --tags "${TAGS//$'\n'/}" \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
    --region us-east-1;
```

Utilizzo dell'SDK AWS per Java

Example

Nell'esempio seguente vengono inseriti i tag di un processo di operazioni in batch S3 tramite la 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);
}
```

Recupero dei tag di processo di un processo S3 Batch Operations

È possibile utilizzare `GetJobTagging` per restituire i tag di un processo S3 Batch Operations. Per maggiori informazioni, consulta i seguenti esempi.

Utilizzo di AWS CLI

Nell'esempio seguente vengono recuperati i tag da un processo di operazioni in batch tramite la AWS CLI.

```
aws \
    s3control get-job-tagging \
    --account-id 123456789012 \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
    --region us-east-1;
```

Utilizzo dell'SDK AWS per Java

Example

Nell'esempio seguente vengono recuperati i tag da un processo di operazioni in batch tramite la AWS SDK for Java.

```
public List<S3Tag> getJobTagging(final AWSS3ControlClient awss3ControlClient,
                                 final String jobId) {
    final GetJobTaggingRequest getJobTaggingRequest = new GetJobTaggingRequest()
            .withJobId(jobId);

    final GetJobTaggingResult getJobTaggingResult =
            awss3ControlClient.getJobTagging(getJobTaggingRequest);

    final List<S3Tag> tags = getJobTaggingResult.getTags();

    return tags;
}
```

Controllo delle autorizzazioni per S3 Batch Operations utilizzando i tag di processo

Per facilitare la gestione dei processi di operazioni in batch S3, è possibile aggiungere tag di processo. Con i tag di processo, è possibile controllare l'accesso ai processi di operazioni in batch e imporre che i tag vengano applicati quando viene creato un processo.

È possibile applicare fino a 50 tag di processo a ciascun processo di operazioni in batch. Ciò consente di impostare policy molto granulari che limitano l'insieme di utenti che possono modificare il lavoro. I tag di lavoro possono favorire o limitare la capacità di un utente di annullare un lavoro, attivare un lavoro in stato di conferma o cambiare il livello di priorità di un lavoro. Inoltre, è possibile applicare i tag a tutti i nuovi lavori e specificare le coppie chiave-valore consentite per i tag. È possibile esprimere tutte queste condizioni utilizzando la stessa lingua delle policy IAM. Per ulteriori informazioni, consulta Azioni, risorse e chiavi di condizione per Amazon S3 nel Service Authorization Reference.

Nell'esempio seguente viene illustrato come utilizzare i tag di processo di operazioni in batch S3 per concedere agli utenti l'autorizzazione per la creazione e la modifica solo dei processi eseguiti all'interno di un reparto specifico (ad esempio, il reparto Finanza o Conformità). È inoltre possibile assegnare i lavori in base allo stadio di sviluppo a cui sono correlati, ad esempio QA o Produzione.

In questo esempio, utilizzi i tag di lavoro di S3 Batch Operations nelle policy AWS Identity and Access Management (IAM) per concedere agli utenti il permesso di creare e modificare solo i job eseguiti all'interno del loro reparto. Assegnare i lavori in base alla fase di sviluppo a cui sono correlati, ad esempio QA o Produzione.

In questo esempio vengono utilizzati i reparti seguenti, ciascuno che utilizza le operazioni in batch in modi diversi:

- Finanza
- Conformità
- Business Intelligence
- Engineering (Progettazione)

Argomenti

- Controllo degli accessi mediante l'assegnazione di tag a utenti e risorse
- Tagging dei processi di operazioni in batch per fase e applicazione dei limiti sulla priorità del processo

Controllo degli accessi mediante l'assegnazione di tag a utenti e risorse

In questo scenario, gli amministratori utilizzano il controllo di accesso basato su attributi (ABAC). ABAC è una strategia di autorizzazione IAM che definisce le autorizzazioni allegando tag sia agli utenti che alle risorse. AWS

Agli utenti e ai lavori viene assegnato uno dei seguenti tag reparto:

Chiave: Valore)

- `department : Finance`
- `department : Compliance`
- `department : BusinessIntelligence`
- `department : Engineering`

> **Note**
>
> I valori e le chiavi dei tag rispettano la distinzione tra maiuscole e minuscole.

Utilizzando la strategia di controllo degli accessi ABAC, concedi a un utente del reparto Finanza l'autorizzazione per la creazione e la gestione dei processi di operazioni in batch Amazon S3 all'interno del proprio reparto associando il tag `department=Finance` al relativo utente.

Inoltre, è possibile collegare una policy gestita all'utente IAM che consente a qualsiasi utente della propria azienda di creare o modificare processi di operazioni in batch Amazon S3 all'interno dei rispettivi reparti.

La policy riportata in questo esempio include tre dichiarazioni di policy:

- La prima istruzione della policy consente all'utente di creare un processo di operazioni in batch a condizione che la richiesta di creazione del processo includa un tag di processo corrispondente al rispettivo reparto. Si esprime utilizzando la sintassi `"${aws:PrincipalTag/department}"`, che viene sostituita dal tag reparto dell'utente al momento della valutazione delle policy. La condizione è soddisfatta quando il valore fornito per il tag reparto nella richiesta (`"aws:RequestTag/department"`) corrisponde al reparto dell'utente.
- La seconda istruzione della policy consente agli utenti di modificare la priorità dei lavori o di aggiornare lo stato di un lavoro a condizione che il lavoro che l'utente sta aggiornando corrisponda al reparto dell'utente.
- La terza istruzione consente a un utente di aggiornare i tag di un processo di operazioni in batch in qualsiasi momento tramite una richiesta `PutJobTagging`, purché (1) il tag del reparto sia conservato e (2) il processo che sta aggiornando sia all'interno del reparto.

```
{
      "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}"
                }
            }
        }
    ]
}
```

Tagging dei processi di operazioni in batch per fase e applicazione dei limiti sulla priorità del processo

Tutti i processi di operazioni in batch Amazon S3 hanno una priorità numerica, che Amazon S3 utilizza per decidere in quale ordine eseguire i processi. In questo esempio, si limita la priorità massima che la maggior parte degli utenti può assegnare ai lavori, con intervalli di priorità più elevati riservati a un gruppo limitato di utenti con privilegi, come segue:

- Intervallo di priorità dello stadio QA (basso): 1-100
- Intervallo di priorità della fase di produzione (alto): 1-300

Per fare ciò, introdurre un nuovo set di tag che rappresenta la fase del lavoro:

Chiave: Valore)

- `stage : QA`
- `stage : Production`

Creazione e aggiornamento di lavori con priorità bassa all'interno di un reparto

Questa policy introduce due nuove restrizioni per la creazione e l'aggiornamento di processi di operazioni in batch S3, oltre alla restrizione basata sul reparto:

- Consente agli utenti di creare o aggiornare lavori nel proprio reparto con una nuova condizione che richiede che il lavoro includa il tag `stage=QA`.
- Consente agli utenti di creare o aggiornare la priorità di un lavoro fino a una nuova priorità massima di 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": "*"
    }
    ]
}
```

Creazione e aggiornamento di lavori ad alta priorità all'interno di un reparto

Un numero ristretto di utenti potrebbe richiedere la possibilità di creare lavori con priorità elevata in QA o Produzione. Per supportare questa esigenza, è possibile creare una policy gestita adattata alla policy con priorità bassa nella sezione precedente.

Questa policy esegue le seguenti operazioni:

- Consente agli utenti di creare o aggiornare lavori nel proprio reparto con il tag `stage=QA` o `stage=Production`.
- Consente agli utenti di creare o aggiornare la priorità di un lavoro fino a un massimo di 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"
                    ]
                }
            }
        }
    ]
}
```

Gestione del blocco oggetti S3 utilizzando S3 Batch Operations

Con il blocco oggetti S3 è possibile inserire anche un blocco di carattere legale sulla versione di un oggetto. Analogamente all'impostazione di un periodo di conservazione, un blocco a di carattere

legale impedisce che una versione di un oggetto venga sovrascritta o eliminata. Tuttavia, un blocco a fini giudiziari non ha un periodo di conservazione associato e rimane valido fino a quando non viene rimosso. Per ulteriori informazioni, consulta Blocco di carattere legale del blocco oggetti S3.

Per informazioni sull'utilizzo di operazioni in batch S3 con blocco oggetti per aggiungere blocchi di carattere legale a molti oggetti Amazon S3 contemporaneamente, consulta le sezioni seguenti.

Argomenti

- Abilitazione del blocco oggetti S3 utilizzando S3 Batch Operations
- Impostazione della conservazione del blocco oggetti mediante Batch Operations
- Utilizzo delle operazioni in batch S3 con la modalità di conformità della conservazione del blocco oggetti S3
- Utilizzare le operazioni in batch S3 con la modalità di governance della conservazione del blocco oggetti S3
- Utilizzo delle operazioni in batch S3 per disattivare il blocco di carattere legale del blocco oggetti S3

Abilitazione del blocco oggetti S3 utilizzando S3 Batch Operations

È possibile utilizzare le operazioni in batch S3 con il blocco oggetti S3 per gestire la conservazione o abilitare un blocco di carattere per molti oggetti Amazon S3 contemporaneamente. Specifica l'elenco degli oggetti di destinazione nel manifest e inviarlo alle operazioni in batch per il completamento. Per ulteriori informazioni, consulta the section called “Conservazione Blocco oggetto” e the section called “Blocco di carattere legale del blocco oggetto”.

Negli esempi seguenti viene illustrato come creare un ruolo IAM con autorizzazioni S3 Batch Operations e aggiornare le autorizzazioni del ruolo per creare processi che abilitano il blocco oggetti. Negli esempi, sostituire qualsiasi valore delle variabili con valori adatti alle proprie esigenze. È inoltre necessario disporre di un manifest `CSV` che identifichi gli oggetti per il processo di operazioni in batch S3. Per ulteriori informazioni, consulta the section called “Specifica di un manifest”.

Usando il AWS CLI

1. Creare un ruolo IAM e assegnare autorizzazioni delle operazioni in batch S3 per l'esecuzione.

   Questa fase è necessaria per tutti i processi di operazioni in batch 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. Configurare le operazioni in batch S3 con il blocco oggetti S3 per l'esecuzione.

   In questa fase è possibile consentire al ruolo di eseguire le operazioni seguenti:

   a. Eseguire il blocco oggetti sul bucket S3 che contiene gli oggetti di destinazione su cui eseguire le operazioni in batch.

   b. Leggere il bucket S3 in cui si trovano il file manifest CSV e gli oggetti.

   c. Scrivere i risultati del processo di operazioni in batch S3 nel bucket di reporting.

```
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}"
```

Utilizzo dell' AWS SDK for Java

Negli esempi seguenti viene illustrato come creare un ruolo IAM con autorizzazioni delle operazioni in batch S3 e aggiornare le autorizzazioni del ruolo per creare processi che abilitano il blocco oggetti tramite la AWS SDK for Java. Nel codice, sostituire qualsiasi valore delle variabili con valori adatti alle proprie esigenze. È inoltre necessario disporre di un manifest CSV che identifichi gli oggetti per il processo di operazioni in batch S3. Per ulteriori informazioni, consulta the section called “Specifica di un manifest”.

Completa la seguente procedura:

1. Creare un ruolo IAM e assegnare autorizzazioni delle operazioni in batch S3 per l'esecuzione. Questa fase è necessaria per tutti i processi di operazioni in batch S3.
2. Configurare le operazioni in batch S3 con il blocco oggetti S3 per l'esecuzione.

   Consenti al ruolo di eseguire le seguenti operazioni:

1. Eseguire il blocco oggetti sul bucket S3 che contiene gli oggetti di destinazione su cui eseguire le operazioni in batch.
2. Leggere il bucket S3 in cui si trovano il file manifest CSV e gli oggetti.
3. Scrivere i risultati del processo di operazioni in batch S3 nel bucket di reporting.

```
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);
}
```

Impostazione della conservazione del blocco oggetti mediante Batch Operations

L'esempio seguente consente alla regola di impostare la conservazione del blocco oggetti S3 per gli oggetti nel bucket manifest.

Aggiorna il ruolo per includere le autorizzazioni `s3:PutObjectRetention` in modo da poter eseguire la conservazione del blocco oggetti sugli oggetti nel bucket.

Usando il 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}"
```

Utilizzo dell' AWS SDK for 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);
}
```

Utilizzo delle operazioni in batch S3 con la modalità di conformità della conservazione del blocco oggetti S3

L'esempio seguente si basa sugli esempi precedenti di creazione di una policy di attendibilità e sull'impostazione delle autorizzazioni di configurazione di S3 Batch Operations e del blocco oggetti S3 per gli oggetti. Questo esempio imposta la modalità di `COMPLIANCE` conservazione `retain until date` al 1° gennaio 2025. Crea un processo che indirizza gli oggetti nel bucket manifest e notifica i risultati nel bucket dei report identificato.

Usando il AWS CLI

Example Impostare la conformità della menzione tra più oggetti

```
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 Estendi la **`COMPLIANCE`** modalità **`retain until date`** fino al 15 gennaio 2025

L'esempio seguente estende la `COMPLIANCE` della modalità `retain until date` al 15 gennaio 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";
```

Utilizzo dell' AWS SDK for Java

Example Imposta la modalità di conservazione su COMPLIANCE e conservala fino alla data del 1° gennaio 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 Estensione della **`COMPLIANCE`** della modalità **`retain until date`**

L'esempio seguente estende la `COMPLIANCE` della modalità `retain until date` al 15 gennaio 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();
}
```

Utilizzare le operazioni in batch S3 con la modalità di governance della conservazione del blocco oggetti S3

L'esempio seguente si basa sull'esempio precedente di creazione di una policy di attendibilità e sull'impostazione delle autorizzazioni di configurazione di operazioni in batch S3 e del blocco oggetti S3. Viene illustrato come applicare la governance del blocco oggetti S3 con `retain until date` al 30 gennaio 2025 su più oggetti. Crea un processo di operazioni in batch che utilizza il bucket manifest e notifica i risultati nel bucket dei report.

Utilizzando il AWS CLI

Example Applica la governance di conservazione di S3 Object Lock su più oggetti con la data di conservazione fino al 30 gennaio 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 Ignorare la governance della conservazione tra più oggetti

L'esempio seguente si basa sull'esempio precedente di creazione di una policy di attendibilità e sull'impostazione delle autorizzazioni di configurazione di operazioni in batch S3 e del blocco oggetti S3. Viene illustrato come ignorare la governance della conservazione tra più oggetti e creare un processo di operazioni in batch che utilizza il bucket manifest e notifica i risultati nel bucket dei report.

```
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";
```

Utilizzo dell' AWS SDK for Java

L'esempio seguente si basa sull'esempio precedente di creazione di una policy di attendibilità e sull'impostazione delle autorizzazioni di configurazione di operazioni in batch S3 e del blocco oggetti S3. Mostra come applicare la governance di conservazione di S3 Object Lock con scadenza `retain until date` al 30 gennaio 2025 su più oggetti. Crea un processo di operazioni in batch che utilizza il bucket manifest e notifica i risultati nel bucket dei report.

Example Applica la governance di conservazione di S3 Object Lock su più oggetti con la data di conservazione fino al 30 gennaio 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 Ignorare la governance della conservazione tra più oggetti

L'esempio seguente si basa sull'esempio precedente di creazione di una policy di attendibilità e sull'impostazione delle autorizzazioni di configurazione di operazioni in batch S3 e del blocco oggetti S3. Viene illustrato come ignorare la governance della conservazione tra più oggetti e creare un processo di operazioni in batch che utilizza il bucket manifest e notifica i risultati nel bucket dei report.

```
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();
}
```

Utilizzo delle operazioni in batch S3 per disattivare il blocco di carattere legale del blocco oggetti S3

L'esempio seguente si basa sugli esempi precedenti di creazione di una policy di attendibilità e sull'impostazione delle autorizzazioni di configurazione di operazioni in batch S3 e del blocco oggetti S3. Viene illustrato come disattivare il blocco legale del blocco oggetti sugli oggetti tramite le operazioni in batch.

Nell'esempio viene innanzitutto aggiornato il ruolo per concedere le autorizzazioni `s3:PutObjectLegalHold`, viene creato un processo di operazioni in batch che disattiva (rimuove) il blocco di carattere legale dagli oggetti identificati nel manifest, quindi viene inviata una segnalazione.

Utilizzando il AWS CLI

Example Aggiorna il ruolo per concedere le autorizzazioni **`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 Disattiva il blocco di carattere legale

Nell'esempio seguente viene disattivato il blocco di carattere legale.

```
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";
```

Utilizzo dell' AWS SDK for Java

Example Aggiorna il ruolo per concedere le autorizzazioni **`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 Disattiva il blocco di carattere legale

Utilizzare l'esempio seguente se si desidera disattivare il blocco di carattere legale.

```
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 sulle operazioni in batch S3

I seguenti tutorial illustrano le procedure complete end-to-end per alcune operazioni in batch.

- Tutorial: transcodifica in batch di video con S3 Batch Operations e AWS LambdaAWS Elemental MediaConvert

Monitoraggio di Amazon S3

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di Amazon S3 e delle tue AWS soluzioni. Ti consigliamo di raccogliere i dati di monitoraggio da tutte le parti della AWS soluzione in modo da poter eseguire più facilmente il debug di un errore multipunto, se si verifica. Prima di iniziare il monitoraggio di Amazon S3, è opportuno creare un piano di monitoraggio che includa le risposte alle seguenti domande:

- Quali sono gli obiettivi del monitoraggio?
- Di quali risorse si intende eseguire il monitoraggio?
- Con quale frequenza sarà eseguito il monitoraggio di queste risorse?
- Quali strumenti di monitoraggio verranno utilizzati?
- Chi eseguirà i processi di monitoraggio?
- Chi deve ricevere una notifica quando si verifica un problema?

Per ulteriori informazioni sulla registrazione e il monitoraggio in Amazon S3, consulta gli argomenti riportati di seguito.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Argomenti

Strumenti di monitoraggio

AWS offre diversi strumenti che puoi utilizzare per monitorare Amazon S3. Alcuni di questi strumenti possono essere configurati in modo che eseguano automaticamente il monitoraggio, mentre altri richiedono l'intervento manuale. Si consiglia di automatizzare il più possibile i processi di monitoraggio.

Strumenti di monitoraggio automatici

Per controllare Amazon S3 e segnalare l'eventuale presenza di problemi, puoi usare gli strumenti di monitoraggio automatici seguenti:

- Amazon CloudWatch Alarms: monitora una singola metrica in un periodo di tempo specificato ed esegui una o più azioni in base al valore della metrica rispetto a una determinata soglia in diversi periodi di tempo. L'azione è una notifica inviata a un argomento di Amazon Simple Notification Service (Amazon SNS) o a una policy di Amazon EC2 Auto Scaling. CloudWatch gli allarmi non richiamano azioni semplicemente perché si trovano in uno stato particolare. Lo stato deve essere cambiato e restare costante per un numero specificato di periodi. Per ulteriori informazioni, consulta Monitoraggio delle metriche con Amazon CloudWatch.
- AWS CloudTrail Monitoraggio dei log: condividi i file di CloudTrail registro tra account, monitora i file di registro in tempo reale inviandoli a CloudWatch Logs, scrivi applicazioni di elaborazione dei log in Java e verifica che i file di registro non siano cambiati dopo la consegna da parte di. CloudTrail Per ulteriori informazioni, consulta Registrazione delle chiamate API Amazon S3 tramite AWS CloudTrail.

Strumenti di monitoraggio manuali

Un'altra parte importante del monitoraggio di Amazon S3 consiste nel monitorare manualmente gli elementi che gli CloudWatch allarmi non coprono. Amazon S3 e altri AWS Management Console dashboard forniscono una at-a-glance visione dello stato del tuo ambiente. CloudWatch Trusted Advisor AWS È possibile abilitare la registrazione degli accessi al server, che monitora le richieste di accesso al bucket. Ogni record del log di accesso contiene i dettagli su una singola richiesta di accesso, ad esempio il richiedente, il nome del bucket, l'ora della richiesta, l'operazione della richiesta, lo stato della risposta e un eventuale codice di errore. Per ulteriori informazioni, consulta Registrazione delle richieste con registrazione dell'accesso al server.

- Il dashboard Amazon S3 mostra quanto segue:

  - I bucket e gli oggetti e le proprietà in essi contenuti.
- La CloudWatch home page mostra quanto segue:
  - Stato e allarmi attuali
  - Grafici degli allarmi e delle risorse
  - Stato di integrità dei servizi

  Inoltre, è possibile utilizzare CloudWatch per effettuare le seguenti operazioni:
  - Creare pannelli di controllo personalizzati per monitorare i servizi di interesse.
  - Creare grafici dei dati dei parametri per la risoluzione di problemi e il rilevamento di tendenze.
  - Cerca e sfoglia tutte le metriche AWS delle tue risorse.
  - Crea e modifica gli allarmi per ricevere le notifiche dei problemi.
- AWS Trusted Advisor può aiutarti a monitorare AWS le tue risorse per migliorare prestazioni, affidabilità, sicurezza ed economicità. Per tutti gli utenti sono disponibili quattro controlli di Trusted Advisor ; per gli utenti con un piano di assistenza Business o Enterprise sono disponibili più di 50 controlli. Per ulteriori informazioni, consulta AWS Trusted Advisor.

  Trusted Advisor dispone dei seguenti controlli relativi ad Amazon S3:
  - Controlli della configurazione di registrazione dei bucket di Amazon S3.
  - Controlli della sicurezza per i bucket di Amazon S3 dotati di autorizzazioni di accesso aperte.
  - Controlli della tolleranza ai guasti per i bucket di Amazon S3 per i quali la funzione Controllo delle versioni non è abilitata o è sospesa.

Opzioni di registrazione per Amazon S3

Puoi registrare le azioni intraprese dagli utenti, dai ruoli o Servizi AWS sulle risorse di Amazon S3 e conservare i record di registro per scopi di controllo e conformità. A tale scopo, è possibile utilizzare i log degli accessi al server, i log AWS CloudTrail o una combinazione di entrambi. Ti consigliamo di utilizzarlo CloudTrail per registrare azioni a livello di bucket e a livello di oggetto per le tue risorse Amazon S3. Per ulteriori informazioni su ciascuna opzione, consulta le sezioni riportate di seguito.

- Registrazione delle richieste con registrazione dell'accesso al server
- Registrazione delle chiamate API Amazon S3 tramite AWS CloudTrail

La tabella seguente elenca le proprietà chiave dei log e dei CloudTrail log di accesso al server Amazon S3. Per assicurarti che CloudTrail soddisfi i tuoi requisiti di sicurezza, consulta la tabella e le note.

Proprietà dei log	AWS CloudTrail	Log di server Amazon S3
Può essere inoltrato ad altri sistemi (Amazon CloudWatch Logs, Amazon Events) CloudWatch	Sì	No
Distribuisce i log a più destinazioni (ad esempio, invia lo stesso log a due diversi bucket)	Sì	No
Attiva i log per un sottoinsieme di oggetti (prefisso)	Sì	No
Distribuzione di log multi-account (bucket di origine e di destinazione di proprietà di account diversi)	Sì	No
Convalida dell'integrità del file di log mediante firma digitale o hashing	Sì	No
Valori predefiniti o scelta della crittografia per i file di log	Sì	No
Operazioni sugli oggetti (mediante le API Amazon S3)	Sì	Sì
Operazioni sui bucket (mediante le API Amazon S3)	Sì	Sì
Interfaccia utente ricercabile per i log	Sì	No

Proprietà dei log	AWS CloudTrail	Log di server Amazon S3
Campi per i parametri di blocco degli oggetti, proprietà Select di Amazon S3 per i record di log	Sì	No
Campi per `Object Size`, `Total Time`, `Turn-Around Time` e `HTTP Referer` per i record di log	No	Sì
Transizioni, scadenze e ripristini del ciclo di vita	No	Sì
Logging delle chiavi in un'operazione di eliminazione batch	No	Sì
Errori di autenticazione[1]	No	Sì
Account in cui vengono distribuiti i log	Proprietario del bucket [2] e richiedente	Solo proprietario del bucket
Performance and Cost	AWS CloudTrail	Amazon S3 Server Logs
Prezzo	Gli eventi di gestione (prima distribuzione) sono gratuiti, gli eventi di dati sono a pagamento, oltre ai log di storage	Nessun costo aggiuntivo oltre all'archiviazione dei log
Velocità di distribuzione dei log	Eventi di dati ogni 5 minuti, eventi di gestione ogni 15 minuti	Entro qualche ora
Formato dei log	JSON	File di log con record delimitati da nuove righe e separati da spazi

Note

1. CloudTrail non fornisce log per le richieste che non superano l'autenticazione (in cui le credenziali fornite non sono valide). Include, tuttavia, i log di richieste in cui l'autenticazione non riesce (`AccessDenied`) e delle richieste effettuate da utenti anonimi.
2. Il proprietario del bucket S3 riceve CloudTrail i log quando l'account non ha accesso completo all'oggetto nella richiesta. Per ulteriori informazioni, consulta Azioni a livello di oggetto di Amazon S3 in scenari tra più account.
3. S3 non supporta l'invio di CloudTrail log o log di accesso al server al richiedente o al proprietario del bucket per le richieste degli endpoint VPC quando la policy dell'endpoint VPC le nega o per le richieste che falliscono prima che la policy VPC venga valutata.

Registrazione delle chiamate API Amazon S3 tramite AWS CloudTrail

Amazon S3 è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o un. Servizio AWS CloudTrail acquisisce tutte le chiamate API per Amazon S3 come eventi. Le chiamate acquisite includono chiamate dalla console Amazon S3 e chiamate in codice alle operazioni dell'API Amazon S3. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare la richiesta effettuata ad Amazon S3, l'indirizzo IP da cui è stata effettuata, quando è stata effettuata e ulteriori dettagli.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali utente root o utente.
- Se la richiesta è stata effettuata per conto di un utente IAM Identity Center.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro Servizio AWS.

CloudTrail è attivo nel tuo account Account AWS quando crei l'account e hai automaticamente accesso alla cronologia degli CloudTrail eventi. La cronologia CloudTrail degli eventi fornisce un record visualizzabile, ricercabile, scaricabile e immutabile degli ultimi 90 giorni di eventi di gestione registrati in un. Regione AWSPer ulteriori informazioni, consulta Lavorare con la cronologia degli

CloudTrail eventi nella Guida per l'utente.AWS CloudTrail Non sono CloudTrail previsti costi per la visualizzazione della cronologia degli eventi.

Per una registrazione continua degli eventi degli Account AWS ultimi 90 giorni, crea un trail o un data store di eventi CloudTrailLake.

CloudTrail sentieri

Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Tutti i percorsi creati utilizzando il AWS Management Console sono multiregionali. È possibile creare un percorso a regione singola o multiregione utilizzando. AWS CLI La creazione di un percorso multiregionale è consigliata in quanto consente di registrare l'intera attività del proprio account Regioni AWS . Se crei un percorso a regione singola, puoi visualizzare solo gli eventi registrati nel percorso. Regione AWS Per ulteriori informazioni sui percorsi, consulta Creazione di un percorso per te Account AWS e Creazione di un percorso per un'organizzazione nella Guida per l'AWS CloudTrail utente.

Puoi inviare gratuitamente una copia dei tuoi eventi di gestione in corso al tuo bucket Amazon S3 CloudTrail creando un percorso, tuttavia ci sono costi di storage di Amazon S3. Per ulteriori informazioni sui CloudTrail prezzi, consulta la pagina Prezzi.AWS CloudTrail Per informazioni sui prezzi di Amazon S3, consulta Prezzi di Amazon S3.

CloudTrail Lake Event Data Store

CloudTrail Lake ti consente di eseguire query basate su SQL sui tuoi eventi. CloudTrail Lake converte gli eventi esistenti in formato JSON basato su righe in formato Apache ORC. ORC è un formato di archiviazione a colonne ottimizzato per il recupero rapido dei dati. Gli eventi vengono aggregati in archivi di dati degli eventi, che sono raccolte di eventi immutabili basate sui criteri selezionati applicando i selettori di eventi avanzati. I selettori applicati a un archivio di dati degli eventi controllano quali eventi persistono e sono disponibili per l'esecuzione della query. Per ulteriori informazioni su CloudTrail Lake, consulta Working with AWS CloudTrail Lake nella Guida per l'utente.AWS CloudTrail

CloudTrail Gli archivi e le richieste di dati sugli eventi di Lake comportano dei costi. Quando crei un datastore di eventi, scegli l'opzione di prezzo da utilizzare per tale datastore. L'opzione di prezzo determina il costo per l'importazione e l'archiviazione degli eventi, nonché il periodo di conservazione predefinito e quello massimo per il datastore di eventi. Per ulteriori informazioni sui CloudTrail prezzi, consulta la sezione Prezzi.AWS CloudTrail

È possibile archiviare i file di log nel bucket per un periodo di tempo indeterminato, ma è anche possibile definire regole per il ciclo di vita di Amazon S3 per archiviare o eliminare automaticamente

i file di log. Per impostazione predefinita, i file di log sono crittografati mediante la crittografia lato server (SSE) di Amazon S3.

Utilizzo CloudTrail dei log con i log di accesso e i log del server Amazon S3 CloudWatch

AWS CloudTrail i log forniscono un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in Amazon S3, mentre i log di accesso al server di Amazon S3 forniscono record dettagliati per le richieste effettuate a un bucket S3. Per ulteriori informazioni sul funzionamento dei diversi log e delle relative proprietà, prestazioni e costi, consulta the section called “Opzioni di registrazione”.

Puoi utilizzare AWS CloudTrail i log insieme ai log di accesso al server per Amazon S3. CloudTrail i log forniscono un monitoraggio dettagliato delle API per le operazioni a livello di bucket e a livello di oggetto di Amazon S3. I log di accesso al server per Amazon S3 ti offrono la visibilità delle operazioni a livello di oggetto sui tuoi dati in Amazon S3. Per ulteriori informazioni sui log degli accessi al server, consultare Registrazione delle richieste con registrazione dell'accesso al server.

Puoi anche utilizzare CloudTrail i log insieme ad Amazon CloudWatch per Amazon S3. CloudTrail l'integrazione con CloudWatch Logs fornisce l'attività dell'API a livello di bucket S3 acquisita da un flusso CloudTrail di CloudWatch log nel gruppo di log specificato CloudWatch . Puoi creare CloudWatch allarmi per monitorare attività API specifiche e ricevere notifiche e-mail quando si verifica un'attività API specifica. Per ulteriori informazioni sugli CloudWatch allarmi per il monitoraggio di attività specifiche dell'API, consulta la Guida per l'AWS CloudTrail utente. Per ulteriori informazioni sull'utilizzo CloudWatch con Amazon S3, consulta. Monitoraggio delle metriche con Amazon CloudWatch

> **Note**
>
> S3 non supporta la consegna dei CloudTrail log al richiedente o al proprietario del bucket per le richieste degli endpoint VPC quando la policy degli endpoint VPC le nega.

CloudTrail tracciamento con chiamate API SOAP di Amazon S3

CloudTrail tiene traccia delle chiamate API SOAP di Amazon S3. Il supporto SOAP di Amazon S3 su HTTP è obsoleto, ma è comunque disponibile su HTTPS. Per ulteriori informazioni sul supporto SOAP di Amazon S3, consulta Appendice A: utilizzo dell'API SOAP.

> **Important**
>
> Le funzioni più recenti di Amazon S3 non sono supportate per SOAP. Ti consigliamo di utilizzare l'API REST o gli AWS SDK.

Azioni SOAP di Amazon S3 tracciate mediante registrazione CloudTrail

Nome API SOAP	Nome dell'evento API utilizzato nel registro CloudTrail
ListAllMyBuckets	`ListBuckets`
CreateBucket	`CreateBucket`
DeleteBucket	`DeleteBucket`
GetBucketAccessControlPolicy	`GetBucketAcl`
SetBucketAccessControlPolicy	`PutBucketAcl`
GetBucketLoggingStatus	`GetBucketLogging`
SetBucketLoggingStatus	`PutBucketLogging`

Per ulteriori informazioni su CloudTrail Amazon S3, consulta i seguenti argomenti:

Argomenti

- Eventi Amazon S3 CloudTrail
- CloudTrail voci dei file di registro per Amazon S3 e S3 su Outposts
- Abilitazione della registrazione CloudTrail degli eventi per bucket e oggetti S3
- Identificazione delle richieste Amazon S3 tramite CloudTrail

Eventi Amazon S3 CloudTrail

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva negli SDK and. AWS Command Line Interface AWS Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

Questa sezione fornisce informazioni sugli eventi a cui S3 effettua l'accesso. CloudTrail

Eventi relativi ai dati di Amazon S3 in CloudTrail

Gli eventi di dati forniscono informazioni sulle operazioni delle risorse eseguite su o in una risorsa (ad esempio, lettura o scrittura su un oggetto Amazon S3). Queste operazioni sono definite anche operazioni del piano dei dati. Gli eventi di dati sono spesso attività che interessano volumi elevati di dati. Per impostazione predefinita, CloudTrail non registra gli eventi relativi ai dati. La cronologia CloudTrail degli eventi non registra gli eventi relativi ai dati.

Per gli eventi di dati sono previsti costi aggiuntivi. Per ulteriori informazioni sui CloudTrail prezzi, consulta la sezione AWS CloudTrail Prezzi.

Puoi registrare gli eventi relativi ai dati per i tipi di risorse Amazon S3 utilizzando la CloudTrail console o AWS CLI le operazioni CloudTrail API. Per ulteriori informazioni su come registrare gli eventi relativi ai dati, consulta Registrazione degli eventi relativi ai dati con AWS Management Console e Registrazione degli eventi relativi ai dati con the AWS Command Line Interface nella Guida per l'utente.AWS CloudTrail

La tabella seguente elenca i tipi di risorse Amazon S3 per i quali è possibile registrare gli eventi relativi ai dati. La colonna Data event type (console) mostra il valore da scegliere dall'elenco Data event type (console) sulla CloudTrail console. La colonna del valore resources.type mostra il `resources.type` valore da specificare durante la configurazione dei selettori di eventi avanzati

utilizzando le API o. AWS CLI CloudTrail La CloudTrail colonna Data API loggate mostra le chiamate API registrate per il tipo di risorsa. CloudTrail

Tipo di evento di dati (console)	valore resources.type	API di dati registrate su 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 di evento di dati (console)	valore resources.type	API di dati registrate su CloudTrail
		• UploadPart • UploadPartCopy

Tipo di evento di dati (console)	valore resources.type	API di dati registrate su CloudTrail
Punto di accesso S3	`AWS::S3::Access Point`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject (solo copie nella stessa regione) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetBucketAcl • GetBucketCors • GetBucketLocation • GetBucketNotificationConfiguration • GetBucketPolicy • GetObject • GetObjectAcl • GetObjectAttributes • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • HeadBucket • HeadObject • ListMultipartUploads • ListObjects • ListObjectsV2 • ListObjectVersions • ListParts • Presign • PutObject

Tipo di evento di dati (console)	valore resources.type	API di dati registrate su CloudTrail
		• PutObjectLegalHold • PutObjectRetention • PutObjectAcl • PutObjectTagging • RestoreObject • UploadPart • UploadPartCopy (solo copie nella stessa regione)

Tipo di evento di dati (console)	valore resources.type	API di dati registrate su CloudTrail
S3 Object Lambda	`AWS::S3ObjectLambda::AccessPoint`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject (solo copie nella stessa regione) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetObject • GetObjectAcl • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • HeadObject • ListMultipartUploads • ListObjects • ListObjectVersions • ListParts • PutObject • PutObjectLegalHold • PutObjectRetention • PutObjectAcl • PutObjectTagging • RestoreObject • UploadPart • WriteGetObjectResponse

Tipo di evento di dati (console)	valore resources.type	API di dati registrate su CloudTrail
S3 Outposts	`AWS::S3Outposts::Object`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject (solo copie nella stessa regione) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetObject • GetObjectTagging • HeadObject • ListMultipartUploads • ListObjects • ListObjectsV2 • ListParts • PutObject • PutObjectTagging • UploadPart • UploadPartCopy

Puoi configurare selettori di eventi avanzati per filtrare in base a `eventName`, `readOnly`, e `resources.ARN` i campi per registrare solo gli eventi che ritieni importanti. Per ulteriori informazioni su questi campi, consulta AdvancedFieldSelectorl'AWS CloudTrail API Reference.

Eventi di gestione di Amazon S3 in CloudTrail

Amazon S3 registra tutte le operazioni del piano di controllo come eventi di gestione. Per ulteriori informazioni sulle operazioni dell'API S3, consulta Amazon S3 API Reference.

Come CloudTrail acquisisce le richieste fatte ad Amazon S3

Per impostazione predefinita, CloudTrail registra le chiamate API a livello di bucket S3 effettuate negli ultimi 90 giorni, ma non registra le richieste fatte agli oggetti. Le chiamate a livello di bucket includono eventi come `CreateBucket`, `DeleteBucket`, `PutBucketLifecycle`, `PutBucketPolicy` e così via. Puoi visualizzare gli eventi a livello di bucket sulla console. CloudTrail Tuttavia, non è possibile visualizzare gli eventi relativi ai dati (chiamate a livello di oggetto Amazon S3): è necessario analizzare o interrogare i log per essi. CloudTrail

Azioni a livello di account Amazon S3 tracciate mediante registrazione CloudTrail

CloudTrail registra le azioni a livello di account. I record Amazon S3 vengono scritti insieme ad altri Servizio AWS record in un file di registro. CloudTrail determina quando creare e scrivere su un nuovo file in base a un periodo di tempo e alle dimensioni del file.

Le tabelle in questa sezione elencano le azioni a livello di account Amazon S3 supportate per la registrazione da. CloudTrail

Le azioni API a livello di account Amazon S3 tracciate tramite CloudTrail registrazione vengono visualizzate con i seguenti nomi di eventi. I nomi degli CloudTrail eventi sono diversi dal nome dell'azione API. Ad esempio, DeletePublicAccessBlock è DeleteAccountPublicAccessBlock.

- DeleteAccountPublicAccessBlock
- GetAccountPublicAccessBlock
- PutAccountPublicAccessBlock

Azioni a livello di bucket di Amazon S3 tracciate mediante registrazione CloudTrail

Per impostazione predefinita, CloudTrail registra le azioni a livello di bucket per i bucket generici. I record Amazon S3 vengono scritti insieme ad altri record di AWS servizio in un file di registro. CloudTrail determina quando creare e scrivere su un nuovo file in base a un periodo di tempo e alle dimensioni del file.

Questa sezione elenca le azioni a livello di bucket di Amazon S3 supportate per la registrazione da. CloudTrail

Le azioni API a livello di bucket di Amazon S3 tracciate tramite CloudTrail registrazione vengono visualizzate con i seguenti nomi di eventi. In alcuni casi, il nome dell' CloudTrail evento è

diverso dal nome dell'azione dell'API. Ad esempio, `PutBucketLifecycleConfiguration` è `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

Oltre a tali operazioni API, è possibile utilizzare anche l'operazione a livello di oggetto oggetto OPTIONS. Questa azione viene considerata come un'azione a livello di bucket nella CloudTrail registrazione perché controlla la configurazione CORS di un bucket.

Azioni a livello di bucket S3 Express One Zone (endpoint API regionale) tracciate mediante registrazione CloudTrail

Per impostazione predefinita, CloudTrail registra le azioni a livello di bucket per i bucket di directory come eventi di gestione. Il formato `eventsource` per gli eventi CloudTrail di gestione per S3 Express One Zone è. `s3express.amazonaws.com`

> **Note**
>
> Per S3 Express One Zone, la CloudTrail registrazione delle operazioni API degli endpoint zonali (a livello di oggetto o piano dati) (ad esempio o) non è supportata. `PutObject` `GetObject`

Le seguenti operazioni Regional Endpoint API vengono registrate su. CloudTrail

- CreateBucket
- DeleteBucket
- DeleteBucketPolicy
- GetBucketPolicy
- PutBucketPolicy
- ListDirectoryBuckets

Per ulteriori informazioni, consulta Best practice per la sicurezza di S3 Express One Zone.

Azioni a livello di oggetto di Amazon S3 in scenari tra più account

Di seguito sono riportati casi d'uso speciali che coinvolgono le chiamate API a livello di oggetto in scenari tra più account e il modo in cui vengono segnalati i log. CloudTrail CloudTrail consegna i log al richiedente (l'account che ha effettuato la chiamata API), tranne in alcuni casi di accesso negato

in cui le voci di registro vengono oscurate o omesse. Quando si imposta l'accesso multiaccount, considerare gli esempi riportati in questa sezione.

> **Note**
>
> Gli esempi presuppongono che i CloudTrail log siano configurati in modo appropriato.

Esempio 1: CloudTrail consegna i log al proprietario del bucket

CloudTrail consegna i log al proprietario del bucket anche se il proprietario del bucket non dispone delle autorizzazioni per la stessa operazione dell'API dell'oggetto. Si consideri il seguente scenario multiaccount:

- L'account A possiede il bucket.
- L'account B (richiedente) tenta di accedere a un oggetto in quel bucket.
- L'account C è proprietario dell'oggetto. L'account C potrebbe essere o non essere lo stesso account dell'account A.

> **Note**
>
> CloudTrail consegna sempre i log delle API a livello di oggetto al richiedente (Account B). CloudTrail Inoltre, fornisce gli stessi log al proprietario del bucket (Account A) anche quando il proprietario del bucket non possiede l'oggetto (Account C) o non dispone delle autorizzazioni per le stesse operazioni API su quell'oggetto.

Esempio 2: non CloudTrail fa proliferare gli indirizzi e-mail utilizzati per impostare gli ACL degli oggetti

Si consideri il seguente scenario multiaccount:

- L'account A possiede il bucket.
- L'account B (richiedente) invia una richiesta per impostare un'assegnazione nell'ACL dell'oggetto utilizzando un indirizzo e-mail. Per ulteriori informazioni sulle ACL, consulta Panoramica delle liste di controllo accessi (ACL).

Il richiedente recupera i log insieme alle informazioni dell'e-mail. Tuttavia, il proprietario del bucket, se è idoneo a ricevere i log, come nell'esempio 1, ottiene il registro che riporta l'evento. CloudTrail

Tuttavia, il proprietario del bucket non riceve le informazioni sulla configurazione dell'ACL, in particolare l'indirizzo e-mail dell'assegnatario e l'assegnazione. L'unica informazione riportata nel log per il proprietario del bucket è che l'account B ha effettuato una chiamata dell'API ACL.

CloudTrail voci dei file di registro per Amazon S3 e S3 su Outposts

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato di crittografia automatico per la configurazione di crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei AWS CloudTrail log, in S3 Inventory, S3 Storage Lens, nella console Amazon S3 e come intestazione di risposta dell'API Amazon S3 aggiuntiva negli SDK and. AWS Command Line Interface AWS Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'operazione dell'API richiesta, la data e l'ora dell'operazione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia stack ordinata delle chiamate API pubbliche, quindi gli eventi non vengono visualizzati in un ordine specifico.

Per maggiori informazioni, consulta i seguenti esempi.

Argomenti

- Esempio: immissione di file di CloudTrail registro per Amazon S3
- Esempio: voci del file di log di Amazon S3 su Outposts

Esempio: immissione di file di CloudTrail registro per Amazon S3

L'esempio seguente mostra una voce di CloudTrail registro che illustra il GETServizio e PutBucketAclGetBucketVersioningle azioni.

```
{
    "Records": [
    {
```

```
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "111122223333",
            "arn": "arn:aws:iam::111122223333:user/myUserName",
            "accountId": "111122223333",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "myUserName"
        },
        "eventTime": "2019-02-01T03:18:19Z",
        "eventSource": "s3.amazonaws.com",
        "eventName": "ListBuckets",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "[]",
        "requestParameters": {
            "host": [
                "s3.us-west-2.amazonaws.com"
            ]
        },
        "responseElements": null,
        "additionalEventData": {
            "SignatureVersion": "SigV2",
            "AuthenticationMethod": "QueryString",
            "aclRequired": "Yes"
    },
        "requestID": "47B8E8D397DCE7A6",
        "eventID": "cdc4b7ed-e171-4cef-975a-ad829d4123e8",
        "eventType": "AwsApiCall",
        "recipientAccountId": "444455556666",
        "tlsDetails": {
            "tlsVersion": "TLSv1.2",
            "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
            "clientProvidedHostHeader": "s3.amazonaws.com"
    }
    },
    {
       "eventVersion": "1.03",
       "userIdentity": {
            "type": "IAMUser",
            "principalId": "111122223333",
            "arn": "arn:aws:iam::111122223333:user/myUserName",
            "accountId": "111122223333",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```
            "userName": "myUserName"
        },
      "eventTime": "2019-02-01T03:22:33Z",
      "eventSource": "s3.amazonaws.com",
      "eventName": "PutBucketAcl",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "",
      "userAgent": "[]",
      "requestParameters": {
          "bucketName": "",
          "AccessControlPolicy": {
              "AccessControlList": {
                  "Grant": {
                      "Grantee": {
                          "xsi:type": "CanonicalUser",
                          "xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
                          "ID":
"d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
                       },
                      "Permission": "FULL_CONTROL"
                   }
              },
              "xmlns": "http://s3.amazonaws.com/doc/2006-03-01/",
              "Owner": {
                  "ID":
"d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
              }
          },
          "host": [
              "s3.us-west-2.amazonaws.com"
          ],
          "acl": [
              ""
          ]
      },
      "responseElements": null,
      "additionalEventData": {
          "SignatureVersion": "SigV4",
          "CipherSuite": "ECDHE-RSA-AES128-SHA",
          "AuthenticationMethod": "AuthHeader"
      },
      "requestID": "BD8798EACDD16751",
      "eventID": "607b9532-1423-41c7-b048-ec2641693c47",
      "eventType": "AwsApiCall",
```

```
      "recipientAccountId": "111122223333",
      "tlsDetails": {
            "tlsVersion": "TLSv1.2",
            "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
            "clientProvidedHostHeader": "s3.amazonaws.com"
    }
    },
    {
      "eventVersion": "1.03",
      "userIdentity": {
          "type": "IAMUser",
          "principalId": "111122223333",
          "arn": "arn:aws:iam::111122223333:user/myUserName",
          "accountId": "111122223333",
          "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
          "userName": "myUserName"
        },
      "eventTime": "2019-02-01T03:26:37Z",
      "eventSource": "s3.amazonaws.com",
      "eventName": "GetBucketVersioning",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "",
      "userAgent": "[]",
      "requestParameters": {
          "host": [
              "s3.us-west-2.amazonaws.com"
          ],
          "bucketName": "example-s3-bucket1",
          "versioning": [
              ""
          ]
      },
      "responseElements": null,
      "additionalEventData": {
          "SignatureVersion": "SigV4",
          "CipherSuite": "ECDHE-RSA-AES128-SHA",
          "AuthenticationMethod": "AuthHeader"
    },
      "requestID": "07D681279BD94AED",
      "eventID": "f2b287f3-0df1-4961-a2f4-c4bdfed47657",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333",
      "tlsDetails": {
            "tlsVersion": "TLSv1.2",
```

```
            "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
            "clientProvidedHostHeader": "s3.amazonaws.com"
    }
    }
  ]
}
```

Esempio: voci del file di log di Amazon S3 su Outposts

Gli eventi di gestione di Amazon S3 on Outposts sono disponibili tramite. AWS CloudTrail Per ulteriori informazioni, consulta Registrazione delle chiamate API Amazon S3 tramite AWS CloudTrail. Inoltre, facoltativamente, puoi abilitare la registrazione per gli eventi di dati in AWS CloudTrail.

Un trail è una configurazione che consente il recapito di eventi come i file di log in un bucket S3 in una regione che specifichi. CloudTrail i registri dei tuoi bucket Outposts includono un nuovo campo `edgeDeviceDetails` che identifica l'Outpost in cui si trova il bucket specificato.

I campi di registro aggiuntivi includono l'azione richiesta, la data e l'ora dell'azione e i parametri della richiesta. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che dimostra un'PutObjectazione su. `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": "example-s3-bucket1",
        "Key": "path/upload.sh"
      },
      "responseElements": {
        "x-amz-server-side-encryption-customer-key-MD5": "wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY",
        "x-amz-server-side-encryption": "Aes256",
        "x-amz-version-id": "001",
        "x-amz-server-side-encryption-customer-algorithm": "Aes256",
        "ETag": "d41d8cd98f00b204e9800998ecf8427f"
      },
      "additionalEventData": {
        "CipherSuite": "ECDHE-RSA-AES128-SHA",
        "bytesTransferredIn": 10,
        "x-amz-id-2": "29xXQBV2O
+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"
    }
```

Abilitazione della registrazione CloudTrail degli eventi per bucket e oggetti S3

Puoi utilizzare gli eventi CloudTrail relativi ai dati per ottenere informazioni sulle richieste a livello di bucket e oggetto in Amazon S3. Per abilitare gli eventi CloudTrail relativi ai dati per tutti i bucket o per un elenco di bucket specifici, devi creare un percorso manualmente in. CloudTrail

> **Note**
>
> - L'impostazione predefinita per CloudTrail è quella di trovare solo gli eventi di gestione. Assicurarsi che gli eventi di dati siano abilitati per l'account.
> - Con un bucket S3 che genera un carico di lavoro elevato, è possibile generare migliaia di log in un breve lasso di tempo. Tieni presente per quanto tempo scegli di abilitare gli eventi CloudTrail relativi ai dati per un bucket occupato.

CloudTrail archivia i log degli eventi dei dati di Amazon S3 in un bucket S3 di tua scelta. Prendi in considerazione l'utilizzo di un bucket separato Account AWS per organizzare meglio gli eventi da più bucket di tua proprietà in un posto centrale per facilitare le interrogazioni e l'analisi. AWS Organizations ti aiuta a crearne uno Account AWS collegato all'account proprietario del bucket che stai monitorando. Per ulteriori informazioni, consulta Cos'è AWS Organizations? nella Guida AWS Organizations per l'utente.

Quando si registrano gli eventi relativi ai dati per un trail in CloudTrail, è possibile scegliere di utilizzare selettori di eventi avanzati o selettori di eventi di base. Quando crei un trail nella CloudTrail console utilizzando selettori di eventi avanzati, nella sezione Data events puoi scegliere Registra tutti gli eventi per il modello Log selector per registrare tutti gli eventi a livello di oggetto. Quando crei un trail nella CloudTrail console utilizzando selettori di eventi di base, nella sezione Data events puoi selezionare la casella di controllo Seleziona tutti i bucket S3 nel tuo account per registrare tutti gli eventi a livello di oggetto.

> **Note**
>
> - È una best practice la creazione di una configurazione del ciclo di vita per il bucket degli eventi di dati AWS CloudTrail . Definisci la configurazione del ciclo di vita in modo tale da rimuovere periodicamente i file di log al termine del periodo di tempo desiderato per l'audit. In questo modo, si riduce la quantità di dati analizzati da Athena per ogni query. Per ulteriori informazioni, consulta Impostazione di una configurazione del ciclo di vita su un bucket.
> - Per ulteriori informazioni sul formato della registrazione, consulta Registrazione delle chiamate API Amazon S3 tramite AWS CloudTrail.
> - Per esempi su come interrogare CloudTrail i log, consulta il post sul blog AWS Big Data Analyze Security, Compliance, and Operational Activity Using AWS CloudTrail and Amazon Athena.

Abilitazione della registrazione per gli oggetti in un bucket utilizzando la console

Puoi utilizzare la console Amazon S3 per configurare un AWS CloudTrail trail per registrare gli eventi relativi ai dati per gli oggetti in un bucket S3. CloudTrail supporta la registrazione di operazioni API a livello di oggetto Amazon S3 come`GetObject`, e. `DeleteObject` `PutObject` Questi eventi vengono chiamati eventi di dati.

Per impostazione predefinita, i CloudTrail trail non registrano gli eventi relativi ai dati, ma puoi configurare i trail per registrare gli eventi di dati per i bucket S3 da te specificati o per registrare gli eventi di dati per tutti i bucket Amazon S3 presenti nel tuo. Account AWS Per ulteriori informazioni, consulta Registrazione delle chiamate API Amazon S3 tramite AWS CloudTrail.

CloudTrail non inserisce gli eventi relativi ai dati nella cronologia degli eventi. CloudTrail Inoltre, non tutte le azioni a livello di bucket sono inserite nella cronologia degli eventi. CloudTrail Per ulteriori

informazioni sulle azioni API a livello di bucket Amazon S3 tracciate mediante registrazione, consulta. CloudTrail Azioni a livello di bucket di Amazon S3 tracciate mediante registrazione CloudTrail Per ulteriori informazioni su come interrogare CloudTrail i log, consulta l'articolo del AWS Knowledge Center sull'uso dei modelli di filtro di Amazon CloudWatch Logs e di Amazon Athena per interrogare i log. CloudTrail

Per configurare un trail per registrare gli eventi di dati per un bucket S3, è possibile utilizzare la console AWS CloudTrail o la console di Amazon S3. Se stai configurando un percorso per registrare gli eventi relativi ai dati per tutti i bucket Amazon S3 presenti nel Account AWS tuo dispositivo, è più facile usare la console. CloudTrail Per informazioni sull'uso della CloudTrail console per configurare un trail per registrare gli eventi relativi ai dati S3, consulta Data events nella Guida per l'utente.AWS CloudTrail

> **Important**
>
> Per gli eventi di dati sono previsti costi aggiuntivi. Per ulteriori informazioni, consulta Prezzi di AWS CloudTrail.

La procedura seguente mostra come utilizzare la console Amazon S3 per configurare un CloudTrail trail per registrare gli eventi relativi ai dati per un bucket S3.

Per abilitare la registrazione degli eventi CloudTrail relativi ai dati per gli oggetti in un bucket S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegliere il nome del bucket.
3. Scegliere Properties (Proprietà).
4. In AWS CloudTrail Data events, scegli Configura in CloudTrail.

   Puoi creare un nuovo CloudTrail percorso o riutilizzare un percorso esistente e configurare gli eventi relativi ai dati di Amazon S3 in modo che vengano registrati nel tuo percorso. Per informazioni su come creare percorsi nella CloudTrail console, consulta Creazione e aggiornamento di un percorso con la console nella Guida per l'utente.AWS CloudTrail Per informazioni su come configurare la registrazione degli eventi dei dati di Amazon S3 nella CloudTrail console, consulta Logging data events for Amazon S3 Objects nella User Guide.AWS CloudTrail

> **Note**
>
> Se utilizzi la CloudTrail console o la console Amazon S3 per configurare un percorso per registrare gli eventi relativi ai dati per un bucket S3, la console Amazon S3 mostra che la registrazione a livello di oggetto è abilitata per il bucket.

Per disabilitare la registrazione degli eventi relativi ai CloudTrail dati per gli oggetti in un bucket S3

1. Accedi AWS Management Console e apri la CloudTrail console all'indirizzo https://console.aws.amazon.com/cloudtrail/.
2. Nel pannello di navigazione a sinistra scegli Trail.
3. Scegli il nome del trail che hai creato per registrare gli eventi del bucket.
4. Nella pagina dei dettagli del trail, scegli Interrompi la registrazione nell'angolo in alto a destra.
5. Nella finestra di dialogo visualizzata, scegli Interrompi la registrazione.

Per informazioni sull'abilitazione della registrazione a livello di oggetto quando si crea un bucket S3, consulta Creazione di un bucket.

Per ulteriori informazioni sulla CloudTrail registrazione con i bucket S3, consulta i seguenti argomenti:

- Visualizzazione delle proprietà di un bucket S3
- Registrazione delle chiamate API Amazon S3 tramite AWS CloudTrail
- Utilizzo dei file di CloudTrail registro nella Guida per l'utente AWS CloudTrail

Identificazione delle richieste Amazon S3 tramite CloudTrail

In Amazon S3, puoi identificare le richieste utilizzando un registro AWS CloudTrail eventi. AWS CloudTrail è il metodo preferito per identificare le richieste Amazon S3, ma se utilizzi i log di accesso al server Amazon S3, consulta. the section called “Identificazione delle richieste S3”

Argomenti

- Identificazione delle richieste effettuate ad Amazon S3 in un registro CloudTrail
- Identificazione delle richieste Amazon S3 Signature versione 2 mediante CloudTrail
- Identificazione dell'accesso agli oggetti S3 utilizzando CloudTrail

Identificazione delle richieste effettuate ad Amazon S3 in un registro CloudTrail

Dopo aver configurato l' CloudTrail invio di eventi a un bucket, dovresti iniziare a vedere gli oggetti andare al bucket di destinazione sulla console Amazon S3. Questi sono formattati come riportato di seguito:

```
s3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/Region/yyyy/mm/dd
```

Gli eventi registrati da CloudTrail vengono archiviati come oggetti gzipped JSON compressi nel bucket S3. Per trovare in modo efficiente le richieste, è necessario utilizzare un servizio come Amazon Athena per indicizzare e interrogare i CloudTrail log.

Per ulteriori informazioni su CloudTrail e Athena, consulta Creazione della tabella per i AWS CloudTrail log in Athena utilizzando la proiezione delle partizioni nella Amazon Athena User Guide.

Identificazione delle richieste Amazon S3 Signature versione 2 mediante CloudTrail

Puoi utilizzare un registro CloudTrail eventi per identificare quale versione di firma API è stata utilizzata per firmare una richiesta in Amazon S3. Questa possibilità è importante perché il supporto di Signature Version 2 sta per essere disattivato perché obsoleto. Dopo, Amazon S3 non accetterà più le richieste che usano Signature Version 2 e tutte le richieste dovranno usare la firma Signature Version 4.

Ti consigliamo vivamente di CloudTrail utilizzarlo per determinare se alcuni dei tuoi flussi di lavoro utilizzano la firma Signature versione 2. Nel caso, correggili aggiornando le librerie e il codice in modo che utilizzino invece Signature Version 4 per evitare qualsiasi impatto sul business.

Per ulteriori informazioni, consulta Annuncio: AWS CloudTrail per Amazon S3 aggiunge nuovi campi per un controllo di sicurezza avanzato. AWS re:Post

> **Note**
>
> CloudTrail gli eventi per Amazon S3 includono la versione della firma nei dettagli della richiesta con il nome chiave di '. `additionalEventData` Per trovare la versione della firma sulle richieste effettuate per oggetti in Amazon S3 come`GET`, e sulle `DELETE` richieste`PUT`, devi abilitare gli eventi relativi ai CloudTrail dati. (Questa funzionalità è disattivata per impostazione predefinita).

AWS CloudTrail è il metodo preferito per identificare le richieste Signature Version 2. Se utilizzi i log degli accessi del server Amazon S3, consulta Identificazione delle richieste di Signature versione 2 tramite i log degli accessi ad Amazon S3.

Argomenti

- Esempi di query Athena per l'identificazione di richieste Amazon S3 Signature versione 2
- Partizionamento dei dati di Signature versione 2

Esempi di query Athena per l'identificazione di richieste Amazon S3 Signature versione 2

Example : seleziona tutti gli eventi Signature Version 2 e stampa solo **`EventTime`**, **`S3_Action`**, **`Request_Parameters`**, **`Region`**, **`SourceIP`** e **`UserAgent`**

Nella query Athena seguente sostituisci *`s3_cloudtrail_events_db.cloudtrail_table`* con i dettagli Athena e aumenta o rimuovi il limite in base alle necessità.

```
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 - Selezionare tutti i richiedenti che inviano traffico di tipo Signature versione 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
```

Partizionamento dei dati di Signature versione 2

Se è necessario eseguire query su una grande quantità di dati, è possibile ridurre i costi e i tempi di esecuzione di Athena creando una tabella partizionata.

Per farlo, creare una nuova tabella con partizioni nel modo seguente.

```
CREATE EXTERNAL TABLE s3_cloudtrail_events_db.cloudtrail_table_partitioned(
     eventversion STRING,
     userIdentity STRUCT<
         type:STRING,
         principalid:STRING,
         arn:STRING,
         accountid:STRING,
         invokedby:STRING,
         accesskeyid:STRING,
         userName:STRING,
      sessioncontext:STRUCT<
                 attributes:STRUCT<
                 mfaauthenticated:STRING,
                 creationdate:STRING>,
                 sessionIssuer:STRUCT<
                 type:STRING,
                 principalId:STRING,
                 arn:STRING,
                 accountId:STRING,
                 userName:STRING>
             >
          >,
     eventTime STRING,
     eventSource STRING,
     eventName STRING,
     awsRegion STRING,
     sourceIpAddress STRING,
     userAgent STRING,
     errorCode STRING,
     errorMessage STRING,
     requestParameters STRING,
     responseElements STRING,
     additionalEventData STRING,
     requestId STRING,
     eventId STRING,
     resources ARRAY<STRUCT<ARN:STRING,accountId: STRING,type:STRING>>,
     eventType STRING,
     apiVersion STRING,
     readOnly STRING,
     recipientAccountId STRING,
     serviceEventDetails STRING,
```

```
        sharedEventID STRING,
        vpcEndpointId STRING
    )
    PARTITIONED BY (region string, year string, month string, day string)
    ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
    STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
    OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
    LOCATION 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/';
```

Quindi creare le partizioni individualmente. Non è possibile ottenere risultati da date che non sono state create.

```
ALTER TABLE s3_cloudtrail_events_db.cloudtrail_table_partitioned ADD
    PARTITION (region= 'us-east-1', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/us-east-1/2019/02/19/'
    PARTITION (region= 'us-west-1', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/us-west-1/2019/02/19/'
    PARTITION (region= 'us-west-2', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/us-west-2/2019/02/19/'
    PARTITION (region= 'ap-southeast-1', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/ap-southeast-1/2019/02/19/'
    PARTITION (region= 'ap-southeast-2', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/ap-southeast-2/2019/02/19/'
    PARTITION (region= 'ap-northeast-1', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/ap-northeast-1/2019/02/19/'
    PARTITION (region= 'eu-west-1', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/eu-west-1/2019/02/19/'
    PARTITION (region= 'sa-east-1', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/sa-east-1/2019/02/19/';
```

È quindi possibile effettuare la richiesta sulla base di queste partizioni e non è necessario caricare l'intero bucket.

```
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
```

Identificazione dell'accesso agli oggetti S3 utilizzando CloudTrail

Puoi utilizzare i registri AWS CloudTrail degli eventi per identificare le richieste di accesso agli oggetti Amazon S3 per eventi ai dati `GetObject` come`DeleteObject`, `PutObject` e, e scoprire ulteriori informazioni su tali richieste.

L'esempio seguente mostra come ottenere tutte le richieste di `PUT` oggetti per Amazon S3 da un registro AWS CloudTrail eventi.

Argomenti

- Esempi di query Athena per l'identificazione di richieste di accesso agli oggetti Amazon S3

Esempi di query Athena per l'identificazione di richieste di accesso agli oggetti Amazon S3

Negli esempi di query Athena seguenti sostituisci *`s3_cloudtrail_events_db.cloudtrail_table`* con i dettagli Athena e modifica l'intervallo della data in base alle necessità.

Example : seleziona tutti gli eventi con richieste **`PUT`** di accesso agli oggetti e stampa solo **`EventTime`**, **`EventSource`**, **`SourceIP`**, **`UserAgent`**, **`BucketName`**, **`object`** e **`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 : seleziona tutti gli eventi con richieste **GET** di accesso agli oggetti e stampa solo **EventTime**, **EventSource**, **SourceIP**, **UserAgent**, **BucketName**, **object** e **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 : seleziona tutti gli eventi anonimi del richiedente per un bucket in un determinato periodo e stampa solo **EventTime**, **EventName**, **EventSource**, **SourceIP**, **UserAgent**, **BucketName**, **UserARN** e **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 : identifica tutte le richieste che richiedono una ACL per l'autorizzazione

L'esempio di query Amazon Athena seguente mostra come identificare tutte le richieste relative ai bucket S3 che richiedono una lista di controllo degli accessi (ACL) per l'autorizzazione. Se la richiesta richiede una ACL per l'autorizzazione, il valore `aclRequired` in `additionalEventData` è `Yes`. Se non sono richieste ACL, `aclRequired` non è presente. È possibile utilizzare queste informazioni per eseguire la migrazione delle autorizzazioni ACL alle policy di bucket appropriate. Dopo aver creato queste policy di bucket, puoi disabilitare le ACL per questi bucket. Per ulteriori informazioni sulla disabilitazione delle ACL, consulta Prerequisiti per la disabilitazione delle 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

- Questi esempi di query possono essere utili anche per il monitoraggio della sicurezza. Puoi rivedere i risultati per le chiamate `PutObject` o `GetObject` da indirizzi IP o richiedenti imprevisti o non autorizzati e per l'identificazione di eventuali richieste anonime ai bucket.
- La query recupera solo le informazioni a partire dall'orario in cui è stata abilitata la registrazione.

Se utilizzi i log di accesso al server Amazon S3, consulta Identificazione delle richieste di accesso agli oggetti tramite i log degli accessi Amazon S3.

Registrazione delle richieste con registrazione dell'accesso al server

La registrazione degli accessi al server fornisce record dettagliati per le richieste che sono effettuate a un bucket. I log di accesso al server sono utili per numerose applicazioni. Ad esempio, le informazioni del log di accesso possono essere utili nei controlli di accesso e di sicurezza. Queste informazioni possono essere utili anche per comprendere la base clienti e la fattura Amazon S3.

> **Note**
>
> I log degli accessi al server non registrano le informazioni sugli errori di reindirizzamento a regioni sbagliate per le regioni lanciate dopo il 20 marzo 2019. Errori di reindirizzamento della regione errata si verificano quando una richiesta per un oggetto o un bucket viene effettuata al di fuori della regione in cui si trova il bucket.

Come si abilita il recapito dei log?

Per abilitare la distribuzione dei log, attenersi alla procedura di base riportata di seguito. Per informazioni dettagliate, vedi Abilitazione della registrazione degli accessi al server Amazon S3.

1. Fornisci il nome del bucket di destinazione (noto anche come bucket target). Questo bucket è dove vuoi che Amazon S3 salvi i log di accesso come oggetti. Sia i bucket di origine che quelli di destinazione devono trovarsi nella stessa Regione AWS e devono appartenere allo stesso account. Il bucket di destinazione non deve avere una configurazione del periodo di conservazione predefinita di S3 Object Lock. Inoltre, nel bucket di destinazione l'opzione di pagamento a carico del cliente non deve essere abilitata.

   I registri possono essere distribuiti a tutti i bucket di cui si è proprietari che si trovano nella stessa regione del bucket di origine, incluso il bucket di origine stesso. Tuttavia, per una gestione più semplice dei log, si consiglia di salvare i log di accesso in un bucket diverso.

   Quando il bucket di origine e il bucket di destinazione sono lo stesso bucket, vengono creati log aggiuntivi per i log che sono scritti nel bucket, il che crea un loop di log infinito. Ciò potrebbe non essere ideale perché potrebbe causare un piccolo incremento di fatturazione dello storage. Inoltre, i registri aggiuntivi relativi ai log potrebbero rendere difficile trovare il log che si sta cercando.

Se scegli di salvare i log degli accessi nel bucket di origine, è consigliabile specificare un prefisso di destinazione (noto anche come prefisso target) per tutte le chiavi degli oggetti del log. Quando specifichi un prefisso, tutti i nomi degli oggetti del log iniziano con una stringa comune, che semplifica l'identificazione degli oggetti del log.

2. (Facoltativo) Assegna un prefisso a tutte le chiavi degli oggetti del log Amazon S3. Il prefisso di destinazione (noto anche come prefisso target) semplifica l'individuazione degli oggetti del log. Se, ad esempio, specifichi il valore di prefisso `logs/`, ogni chiave degli oggetti del log creato da Amazon S3 è preceduta dal prefisso `logs/`.

   ```
   logs/2013-11-01-21-32-16-E568B2907131C0C0
   ```

   Se specifichi il valore del prefisso `logs`, l'oggetto del log viene visualizzato come segue:

   ```
   logs2013-11-01-21-32-16-E568B2907131C0C0
   ```

   I prefissi sono utili anche per distinguere tra i bucket di origine quando più bucket si collegano allo stesso bucket di destinazione.

   Il prefisso può essere utile nell'eliminazione dei log. Ad esempio, è possibile impostare una regola di configurazione del ciclo di vita per Amazon S3 per eliminare gli oggetti con un prefisso specifico. Per ulteriori informazioni, consulta Eliminazione dei file di log Amazon S3.

3. (Facoltativo) Imposta autorizzazioni che consentano ad altri utenti di accedere ai log generati. Per default, solo il proprietario del bucket dispone di tutte le autorizzazioni per accedere agli oggetti del log. Se il bucket di destinazione utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto S3 per disabilitare le liste di controllo degli accessi (ACL), non potrai approvare le autorizzazioni in concessioni di destinazione (note anche come concessioni target) che utilizzano ACL. Tuttavia, puoi aggiornare la policy di bucket per il bucket di destinazione per concedere l'accesso ad altri utenti. Per ulteriori informazioni, consulta Identity and Access Management per Amazon S3 e Autorizzazioni per la distribuzione dei registri.

4. (Facoltativo) Imposta un formato della chiave dell'oggetto di log per i file di log. Sono disponibili due opzioni per il formato della chiave dell'oggetto di log (noto anche come formato chiave dell'oggetto target):

   - Nessun on-date-based partizionamento: questo è il formato originale della chiave dell'oggetto di registro. Se scegli questo formato, il formato della chiave del file di log viene visualizzato come segue:

```
[DestinationPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Ad esempio, se specifichi `logs/` come prefisso, gli oggetti di log vengono denominati in questo modo:

```
logs/2013-11-01-21-32-16-E568B2907131C0C0
```

- Partizionamento basato sulla data: se scegli il partizionamento basato sulla data, puoi scegliere l'ora dell'evento o l'ora di consegna per il file di log come origine della data utilizzata nel formato di log. Questo formato semplifica l'interrogazione dei log.

  Se scegli il partizionamento basato sulla data, il formato chiave del file di log viene visualizzato come segue:

  ```
  [DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/
  [DD]/[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
  ```

  Ad esempio, se specifichi `logs/` come prefisso target, gli oggetti del log vengono denominati in questo modo:

  ```
  logs/123456789012/us-west-2/DOC-EXAMPLE-SOURCE-
  BUCKET/2023/03/01/2023-03-01-21-32-16-E568B2907131C0C0
  ```

  Per l'ora di consegna, l'ora indicato nei nomi dei file di log corrisponde all'ora di consegna dei file di log.

  Per quanto riguarda l'ora di consegna degli eventi, l'anno, il mese e il giorno corrispondono al giorno in cui si è verificato l'evento e l'ora, i minuti e i secondi sono impostati su `00` nella chiave. I log forniti in questi file di log riguardano solo un giorno specifico.

Se configuri i log tramite AWS Command Line Interface (AWS CLI), AWS SDK o l'API REST di Amazon S3, usali `TargetObjectKeyFormat` per specificare il formato della chiave dell'oggetto di registro. Per specificare non-date-based il partizionamento, usa. `SimplePrefix` Per specificare un partizionamento basato sulla data, utilizza `PartitionedPrefix`. Se si utilizza `PartitionedPrefix`, utilizza `PartitionDateSource` per specificare `EventTime` o `DeliveryTime`.

Infatti `SimplePrefix`, il formato della chiave del file di log è il seguente:

```
[TargetPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Per `PartitionedPrefix` con l'ora dell'evento o l'ora di consegna, il formato della chiave del file di log viene visualizzato come segue:

```
[TargetPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/
[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Formato della chiave dell'oggetto di log

Amazon S3 utilizza i formati della chiave dell'oggetto seguenti per gli oggetti di log che carica nel bucket di destinazione:

- on-date-based Partizionamento N: questo è il formato originale della chiave dell'oggetto di registro. Se scegli questo formato, il formato della chiave del file di log viene visualizzato come segue:

  ```
  [DestinationPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
  ```

- Partizionamento basato sulla data: se scegli il partizionamento basato sulla data, puoi scegliere l'ora dell'evento o l'ora di consegna per il file di log come origine della data utilizzata nel formato di log. Questo formato semplifica l'interrogazione dei log.

  Se scegli il partizionamento basato sulla data, il formato chiave del file di log viene visualizzato come segue:

  ```
  [DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/
  [YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
  ```

Nella chiave dell'oggetto di log, `YYYY`, `MM`, `DD`, `hh`, `mm` e `ss` indicano rispettivamente anno, mese, giorno, ora, minuti e secondi in cui è stato distribuito il file di log. Date e ore sono in formato UTC.

Un file di log distribuito in un orario specifico può contenere report scritti in un momento qualsiasi prima di quell'orario. Non esiste modo di sapere se tutti i report del log per un determinato intervallo di tempo sono stati distribuiti o meno.

Il componente `UniqueString` della chiave serve a impedire che i file vengano sovrascritti. Non ha alcun significato e il software di elaborazione dei log dovrebbe ignorarlo.

Come vengono distribuiti i log?

Amazon S3 raccoglie periodicamente i record dei log degli accessi, li consolida in file di log e quindi carica i file di log nel bucket di destinazione come oggetti log. Se abiliti la registrazione di log in più bucket di origine che identificano lo stesso bucket di destinazione, nel bucket di destinazione saranno presenti i log degli accessi per tutti i bucket di origine. Ogni oggetto del log, tuttavia, fornisce i report del log di accesso per uno specifico bucket di origine.

Amazon S3 utilizza uno speciale account di recapito di registri per scrivere i registri degli accessi nel server. Queste scritture sono soggette alle normali restrizioni del controllo accessi. Si consiglia di aggiornare la policy del bucket nel bucket di destinazione per concedere l'accesso al principale del servizio di registrazione di log (`logging.s3.amazonaws.com`) per la consegna di log degli accessi. Puoi anche concedere l'accesso per la consegna di log degli accessi al gruppo di consegna di log S3 tramite la lista di controllo degli accessi (ACL) del bucket. Tuttavia, non è consigliabile concedere l'accesso al gruppo di consegna di log S3 tramite le ACL del bucket.

Quando abiliti la registrazione degli accessi al server e si concede l'accesso per la consegna di log di accesso tramite la policy del bucket di destinazione, devi aggiornare la policy per consentire l'accesso `s3:PutObject` al principale del servizio di registrazione dei log. Se utilizzi la console di Amazon S3 per abilitare la registrazione dei log degli accessi al server, la console aggiorna automaticamente la policy del bucket di destinazione per concedere tali autorizzazioni al principale del servizio di registrazione di log. Per ulteriori informazioni sulla concessione di autorizzazioni per il recapito del registro degli accessi al server, consulta Autorizzazioni per la distribuzione dei registri.

> **Note**
>
> S3 non supporta l'invio di CloudTrail log o log di accesso al server al richiedente o al proprietario del bucket per le richieste degli endpoint VPC quando la policy dell'endpoint VPC le nega o per le richieste che falliscono prima che la policy VPC venga valutata.

Impostazione proprietario del bucket applicato per S3 Object Ownership

Se il bucket di destinazione utilizza l'impostazione proprietario del bucket applicato per Proprietà dell'oggetto, le ACL vengono disabilitate e non influiscono più sulle autorizzazioni. È necessario aggiornare la policy del bucket nel bucket di destinazione per concedere l'accesso al principale del

servizio di registrazione di log. Per ulteriori informazioni su Object Ownership, consulta Concedere l'accesso al gruppo di consegna di log S3 per la registrazione di log degli accessi al server.

Consegna di log del server sulla base del miglior tentativo

I record dei log degli accessi al server vengono distribuiti sulla base del miglior tentativo. La maggior parte delle richieste di un bucket correttamente configurato per la registrazione determinano la consegna di un report del log. La maggior parte dei record vengono distribuiti entro qualche ora dal momento della creazione, ma possono essere distribuiti come maggiore frequenza.

La completezza e la tempestività della registrazione del server non è tuttavia garantita. È possibile che il record del log per una richiesta specifica venga consegnato molto tempo dopo l'elaborazione effettiva della richiesta o non venga consegnato affatto. Potresti persino notare la duplicazione di un record di log. Lo scopo dei log del server è fornire un'idea della natura del traffico nel bucket. Sebbene sia raro perdere i record di log o vedere la duplicazione di un record di log, tieni presente che la registrazione di log del server non intende essere un resoconto completo di tutte le richieste.

Data la natura basata sul miglior tentativo possibile della registrazione di log del server, i report di utilizzo potrebbero includere una o più richieste di accesso che non appaiono in un log del server consegnato. Puoi trovare questi report di utilizzo in Report costi e utilizzo nella console AWS Billing and Cost Management .

Tempo richiesto per l'applicazione delle modifiche dello stato di registrazione del bucket

L'applicazione effettiva delle modifiche dello stato di registrazione di un bucket sulla distribuzione dei file di log richiede tempo. Ad esempio, se abiliti la registrazione per un bucket, è possibile che nell'ora successiva alcune richieste vengano registrate nel log e altre no. Supponi di cambiare il bucket di destinazione per la registrazione di log dal bucket A al bucket B. Nell'ora successiva, alcuni log potrebbero continuare a essere distribuiti nel bucket A, mentre potrebbero essere consegnati nel nuovo bucket di destinazione, B. In tutti i casi, le nuove impostazioni diventano effettive senza bisogno di ulteriori azioni da parte dell'utente.

Per ulteriori informazioni su registrazione e file di log, consulta le seguenti sezioni:

Argomenti

- Abilitazione della registrazione degli accessi al server Amazon S3
- Formato del log di accesso al server Amazon S3

- Eliminazione dei file di log Amazon S3
- Utilizzo dei log degli accessi al server Amazon S3 per identificare le richieste

Abilitazione della registrazione degli accessi al server Amazon S3

La registrazione degli accessi al server fornisce record dettagliati per le richieste che sono effettuate a un bucket Amazon S3. I log di accesso al server sono utili per numerose applicazioni. Ad esempio, le informazioni del log di accesso possono essere utili nei controlli di accesso e di sicurezza. Queste informazioni possono essere utili anche per comprendere la base clienti e la fattura Amazon S3.

Per default, Amazon S3 non raccoglie i log degli accessi al server. Quando abiliti la registrazione di log, Amazon S3 fornisce i log degli accessi per un bucket di origine a un bucket di destinazione scelto (noto anche come bucket target). Il bucket di destinazione si deve trovare nella stessa Regione AWS e nello stesso Account AWS del bucket di origine.

Un record di log degli accessi contiene informazioni dettagliate sulle richieste effettuate a un bucket, tra cui il tipo di richiesta, le risorse specificate nella richiesta, nonché l'ora e la data di elaborazione della richiesta. Per ulteriori informazioni sui principi di base della registrazione, consulta Registrazione delle richieste con registrazione dell'accesso al server.

Important

- L'abilitazione della registrazione degli accessi al server per un bucket Amazon S3 non prevede addebiti aggiuntivi. Tuttavia, i file di log distribuiti dal sistema accumulano i consueti addebiti per lo storage. È possibile eliminare i file di log in qualsiasi momento. Il costo di trasferimento dei dati per la consegna dei file di log non viene valutato, ma viene addebitata la normale tariffa di trasferimento dei dati per l'accesso ai file di log.
- Il bucket di destinazione non deve avere la registrazione di log degli accessi al server abilitata. I registri possono essere distribuiti a tutti i bucket di cui si è proprietari che si trovano nella stessa regione del bucket di origine, incluso il bucket di origine stesso. Tuttavia, la distribuzione dei log nel bucket di origine causa un ciclo infinito di log e non è consigliata. Tuttavia, per una gestione più semplice dei log, si consiglia di salvare i log di accesso in un bucket diverso. Per ulteriori informazioni, consulta Come si abilita il recapito dei log?
- I bucket S3 con S3 Object Lock abilitato non possono essere utilizzati come bucket di destinazione per i log degli accessi al server. Il bucket di destinazione non deve avere una configurazione del periodo di conservazione predefinita.

- Il bucket di destinazione non deve avere l'opzione di pagamento a carico del cliente abilitata.
- Puoi utilizzare la crittografia bucket predefinita nel bucket di destinazione solo se utilizzi la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3), che utilizza Advanced Encryption Standard a 256 bit (AES-256). La crittografia predefinita lato server con chiavi () (SSE-KMS) non è supportata. AWS Key Management Service AWS KMS

Puoi abilitare o disabilitare la registrazione degli accessi al server utilizzando la console di Amazon S3, l'API Amazon S3, la AWS Command Line Interface (AWS CLI) o gli SDK AWS .

Autorizzazioni per la distribuzione dei registri

Amazon S3 utilizza uno speciale account di recapito dei registri per scrivere i registri degli accessi nel server. Queste scritture sono soggette alle normali restrizioni del controllo accessi. Per la consegna di log degli accessi, è necessario concedere al principale (`logging.s3.amazonaws.com`) del servizio di registrazione di log l'accesso al bucket di destinazione.

Per concedere le autorizzazioni ad Amazon S3 per la consegna di log, è possibile utilizzare una policy del bucket o le liste di controllo degli accessi (ACL) del bucket, a seconda delle impostazioni di Proprietà dell'oggetto S3 del bucket di destinazione. Invece di una lista di controllo degli accessi (ACL) consigliamo di utilizzare una policy del bucket.

Impostazione proprietario del bucket applicato per S3 Object Ownership

Se il bucket di destinazione utilizza l'impostazione proprietario del bucket applicato per Proprietà dell'oggetto, le ACL vengono disabilitate e non influiscono più sulle autorizzazioni. In questo caso, è necessario aggiornare la policy del bucket nel bucket di destinazione per concedere l'accesso al principale del servizio di registrazione di log. Non è possibile aggiornare l'ACL del bucket per concedere l'accesso al gruppo di consegna di log S3. Inoltre, non puoi includere concessioni di destinazione (note anche come concessioni target) nella configurazione PutBucketLogging.

Per informazioni sulla migrazione delle ACL bucket esistenti per la distribuzione di log di accesso a una policy di bucket, consulta Concedere l'accesso al gruppo di consegna di log S3 per la registrazione di log degli accessi al server. Per ulteriori informazioni su Object Ownership, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket. Quando crei nuovi bucket, le ACL sono disabilitate per impostazione predefinita.

Concessione dell'accesso utilizzando una policy del bucket

Per concedere l'accesso utilizzando la policy del bucket nel bucket di destinazione, aggiorna la policy del bucket per concedere l'autorizzazione `s3:PutObject` al principale del servizio di registrazione di log. Se utilizzi la console di Amazon S3 per abilitare la registrazione di log degli accessi al server, la console aggiorna automaticamente la policy nel bucket di destinazione per concedere tali autorizzazioni al principale del servizio di registrazione di log. Se abiliti la registrazione di log degli accessi al server a livello di programmazione, puoi aggiornare manualmente la policy del bucket per il bucket di destinazione per concedere l'accesso al principale del servizio di registrazione di log.

Per un esempio di policy del bucket che concede l'accesso al principale del servizio di registrazione di log, consulta the section called “Concedi le autorizzazioni al principale del servizio di registrazione di log utilizzando una policy del bucket”.

Concessione dell'accesso utilizzando le liste di controllo degli accessi (ACL) del bucket

Puoi utilizzare in alternativa le liste di controllo degli accessi (ACL) del bucket per concedere l'accesso per la consegna di log degli accessi. Aggiungi una voce apposita nell'ACL di bucket che conceda autorizzazioni `WRITE` e `READ_ACP` al gruppo di distribuzione di registri S3. Tuttavia, non è consigliabile concedere l'accesso al gruppo di distribuzione dei log S3 tramite le ACL bucket. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket. Per informazioni sulla migrazione delle ACL bucket esistenti per la distribuzione di log di accesso a una policy di bucket, consulta Concedere l'accesso al gruppo di consegna di log S3 per la registrazione di log degli accessi al server. Per un esempio di ACL che concede l'accesso al principale del servizio di registrazione di log, consulta the section called “Concedere autorizzazioni al gruppo di distribuzione dei log utilizzando l'ACL bucket”.

Concedi le autorizzazioni al principale del servizio di registrazione di log utilizzando una policy del bucket

Questo esempio di policy del bucket concede autorizzazioni `s3:PutObject` al principale del servizio di registrazione di log (`logging.s3.amazonaws.com`). Per utilizzare questa policy del bucket, sostituisci *`user input placeholders`* con le tue informazioni. Nella seguente politica, *`example-s3-destination-bucket`* è il bucket di destinazione in cui verranno consegnati i log di accesso al server ed è il bucket di origine. *`example-s3-source-bucket`* *`EXAMPLE-LOGGING-PREFIX`*è il prefisso di destinazione opzionale (noto anche come prefisso di destinazione) che si desidera utilizzare per gli oggetti di registro. *`SOURCE-ACCOUNT-ID`*è il proprietario del Account AWS bucket di origine.

> **Note**
>
> Se nella policy del bucket sono presenti istruzioni `Deny`, assicurati che non impediscano ad Amazon S3 di distribuire i log di accesso.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ServerAccessLogsPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::example-s3-destination-bucket/EXAMPLE-LOGGING-
PREFIX*",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::example-s3-source-bucket"
                },
                "StringEquals": {
                    "aws:SourceAccount": "SOURCE-ACCOUNT-ID"
                }
            }
        }
    ]
}
```

Concedere autorizzazioni al gruppo di distribuzione dei log utilizzando l'ACL bucket

> **Note**
>
> Come best practice di sicurezza, Amazon S3 disabilita le liste di controllo degli accessi (ACL) per impostazione predefinita in tutti i nuovi bucket. Per ulteriori informazioni sulle autorizzazioni ACL nella console di Amazon S3, consulta Configurazione delle ACL.

Sebbene questo approccio non sia consigliato, puoi concedere le autorizzazioni al gruppo di consegna di log utilizzando una ACL di bucket. Tuttavia, se il bucket di destinazione utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto, non puoi impostare le ACL di bucket o di oggetti. Inoltre, non puoi includere concessioni di destinazione (note anche come concessioni target) nella configurazione PutBucketLogging. Invece, utilizza una policy del bucket per concedere l'accesso al principale del servizio di registrazione (`logging.s3.amazonaws.com`). Per ulteriori informazioni, consulta Autorizzazioni per la distribuzione dei registri.

Nelle liste di controllo degli accessi del bucket, il gruppo di consegna di log è rappresentato dall'URL seguente.

```
http://acs.amazonaws.com/groups/s3/LogDelivery
```

Per concedere le autorizzazioni `WRITE` e `READ_ACP` (lettura ACL), aggiungi le seguenti concessioni all'ACL di bucket di destinazione:

```
<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>
```

Per esempi sull'aggiunta a livello di programmazione di concessioni ACL, consulta la sezione Configurazione delle ACL.

> **Important**
>
> Quando abiliti la registrazione degli accessi al server Amazon S3 utilizzando AWS CloudFormation su un bucket e utilizzi gli ACL per concedere l'accesso al gruppo di consegna dei log di S3, devi anche aggiungere "al tuo modello. `AccessControl": "LogDeliveryWrite"` CloudFormation Questa operazione è importante perché è possibile concedere tali autorizzazioni solo creando un ACL per il bucket, ma non è possibile creare

ACL personalizzati per i bucket in cui si trovano. CloudFormation È possibile utilizzare solo gli ACL predefiniti con. CloudFormation

Come abilitare la registrazione degli accessi al server

Per abilitare la registrazione degli accessi al server utilizzando la console Amazon S3, l'API REST di Amazon S3, gli SDK AWS CLI e AWS , utilizza le seguenti procedure.

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegliere il nome del bucket per il quale si desidera abilitare la registrazione degli accessi al server.
3. Scegliere Properties (Proprietà).
4. Nella sezione Server access logging (Registrazione degli accessi al server) scegliere Edit (Modifica).
5. In Registrazione degli accessi al server, seleziona Abilita.
6. In Bucket di destinazione, specifica un bucket e un prefisso opzionale. Se specifichi un prefisso, ti consigliamo di includere una barra in avanti (/) dopo il prefisso per facilitare la ricerca dei log.

   Note

   L'indicazione di un prefisso con una barra (/) semplifica l'individuazione degli oggetti del log. Se, ad esempio, specifichi il valore di prefisso `logs/`, la chiave di ogni oggetto del log creato da Amazon S3 è preceduta dal prefisso `logs/`, come segue:

   ```
   logs/2013-11-01-21-32-16-E568B2907131C0C0
   ```

   Se specifichi il valore del prefisso `logs`, l'oggetto del log viene visualizzato come segue:

   ```
   logs2013-11-01-21-32-16-E568B2907131C0C0
   ```

7. In Formato della chiave dell'oggetto di log, esegui una delle seguenti operazioni:

- Per scegliere il non-date-based partizionamento, scegli [DestinationPrefix] [YYYY] - [MM] - [DD] - [hh] - [mm] - [ss] - []. UniqueString
- Per scegliere il partizionamento basato sulla data, scegli [DestinationPrefix] [SourceAccountId]/[SourceRegionSourceBucket]/[YYYY]/[MM]/[DD]/[YYYY] - [MM] - [DD] - [hh] - [mm] - [ss] - [UniqueString], quindi scegli S3 event time o Log file delivery time.

8. Seleziona Salvataggio delle modifiche.

   Quando abiliti la registrazione di log degli accessi al server in un bucket, la console abilita la registrazione nel bucket di origine e aggiorna la policy del bucket per il bucket di destinazione in modo da concedere autorizzazioni `s3:PutObject` al principale del servizio di registrazione di log (`logging.s3.amazonaws.com`). Per ulteriori informazioni su questa policy del bucket, consulta Concedi le autorizzazioni al principale del servizio di registrazione di log utilizzando una policy del bucket.

   Puoi visualizzare i log nel bucket di destinazione. Dopo aver abilitato la registrazione degli accessi al server, potrebbero essere necessarie ore prima che i log vengano consegnati al bucket di destinazione. Per ulteriori informazioni su come e quando vengono distribuiti i log, consultare Come vengono distribuiti i log?.

Per ulteriori informazioni, consulta Visualizzazione delle proprietà di un bucket S3.

Utilizzo di REST API

Per abilitare i log, devi inviare una richiesta PutBucketLogging per aggiungere la configurazione della registrazione di log nel bucket di origine. La richiesta specifica il bucket di destinazione (noto anche come bucket target) e, facoltativamente, il prefisso da utilizzare con tutte le chiavi degli oggetti del log.

L'esempio seguente identifica *`example-s3-destination-bucket`* come bucket di destinazione e *`logs/`* come prefisso.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>example-s3-destination-bucket</TargetBucket>
    <TargetPrefix>logs/</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>
```

L'esempio seguente identifica *example-s3-destination-bucket* come bucket di destinazione, *logs/* come prefisso e `EventTime` come il formato della chiave dell'oggetto di log.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>example-s3-destination-bucket</TargetBucket>
    <TargetPrefix>logs/</TargetPrefix>
    <TargetObjectKeyFormat>
      <PartitionedPrefix>
         <PartitionDateSource>EventTime</PartitionDateSource>
      </PartitionedPrefix>
  </TargetObjectKeyFormat>
  </LoggingEnabled>
</BucketLoggingStatus>
```

Gli oggetti del registri vengono scritti dall'account di distribuzione di log S3 e sono di proprietà di tale account. Al proprietario del bucket vengono concesse autorizzazioni complete sugli oggetti del log. Puoi usare in modo opzionale le concessioni di destinazione (note anche come concessioni target) per concedere le autorizzazioni ad altri utenti in modo che possano accedere ai log. Per ulteriori informazioni, consulta PutBucketLogging.

> **Note**
>
> Se il bucket di destinazione utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto, non puoi utilizzare le concessioni di destinazione per assegnare autorizzazioni ad altri utenti. Per concedere autorizzazioni ad altri utenti, puoi aggiornare la policy del bucket nel bucket di destinazione. Per ulteriori informazioni, consulta Autorizzazioni per la distribuzione dei registri.

Per recuperare la configurazione di registrazione di log su un bucket, utilizza l'operazione API GetBucketLogging.

Per eliminare la configurazione della registrazione di log, devi inviare una richiesta `PutBucketLoggingcon` un elemento `BucketLoggingStatus` vuoto:

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
</BucketLoggingStatus>
```

Per abilitare la registrazione su un bucket, puoi utilizzare l'API Amazon S3 o le librerie wrapper SDK AWS .

Utilizzo degli SDK AWS

Gli esempi seguenti abilitano la registrazione di log in un bucket. Devi creare due bucket, uno di origine e uno di destinazione (target). Gli esempi aggiornano prima l'ACL del bucket sul bucket di destinazione. Quindi concedono al gruppo di consegna di log le autorizzazioni necessarie per scrivere i log sul bucket di destinazione e poi abilitano la registrazione di log sul bucket di origine.

Questi esempi non funzionano sui bucket di destinazione che utilizzano l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto.

Se il bucket di destinazione (target) utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto, non sarà possibile impostare le ACL di bucket o di oggetti. Inoltre, non puoi includere sovvenzioni di destinazione (target) nella tua PutBucketLoggingconfigurazione. Occorre utilizzare una policy di bucket per concedere le autorizzazioni di accesso al principal del servizio di registrazione (`logging.s3.amazonaws.com`). Per ulteriori informazioni, consulta Autorizzazioni per la distribuzione dei registri.

.NET

AWS SDK for .NET

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
        }
    }
```

- Per i dettagli sull'API, PutBucketLoggingconsulta AWS SDK for .NET API Reference.

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());
    }

}
```

Utilizzando il AWS CLI

Ti consigliamo di creare un bucket di registrazione dedicato in ogni bucket S3 in Regione AWS cui sono presenti bucket S3. Quindi, fare in modo che i log degli accessi Amazon S3 vengano recapitati al bucket S3. Per ulteriori informazioni ed esempi, consulta l'esempio put-bucket-logging in Riferimento ai comandi della AWS CLI .

Se il bucket di destinazione (target) utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto, non sarà possibile impostare le ACL di bucket o di oggetti. Inoltre, non puoi includere sovvenzioni di destinazione (target) nella tua configurazione. PutBucketLogging Occorre utilizzare una policy di bucket per concedere le autorizzazioni di accesso al principal del servizio di registrazione (`logging.s3.amazonaws.com`). Per ulteriori informazioni, consulta Autorizzazioni per la distribuzione dei registri.

Example - Abilitare i log degli accessi con cinque bucket in due regioni

In questo esempio, sono presenti i cinque bucket seguenti:

- `1-DOC-EXAMPLE-BUCKET1-us-east-1`
- `2-DOC-EXAMPLE-BUCKET1-us-east-1`
- `3-DOC-EXAMPLE-BUCKET1-us-east-1`
- `1-DOC-EXAMPLE-BUCKET1-us-west-2`
- `2-DOC-EXAMPLE-BUCKET1-us-west-2`

> **Note**
>
> Il passaggio finale della procedura seguente fornisce esempi di script bash che è possibile utilizzare per creare i bucket di registrazione di log e abilitare la registrazione di log degli accessi al server su questi bucket. Per utilizzare questi script, devi creare i file `policy.json` e `logging.json`, come descritto nella procedura seguente.

1. Crea due bucket di destinazione per la registrazione di log nelle regioni Stati Uniti occidentali (Oregon) e Stati Uniti orientali (N. Virginia) e assegna loro i nomi elencati di seguito:

   - `DOC-EXAMPLE-BUCKET1-logs-us-east-1`
   - `DOC-EXAMPLE-BUCKET1-logs-us-west-2`

2. Più avanti in questi passaggi, abiliterai la registrazione di log degli accessi al server come segue:

   - `1-DOC-EXAMPLE-BUCKET1-us-east-1` accede al bucket S3 `DOC-EXAMPLE-BUCKET1-logs-us-east-1` con prefisso `1-DOC-EXAMPLE-BUCKET1-us-east-1`
   - `2-DOC-EXAMPLE-BUCKET1-us-east-1` accede al bucket S3 `DOC-EXAMPLE-BUCKET1-logs-us-east-1` con prefisso `2-DOC-EXAMPLE-BUCKET1-us-east-1`
   - `3-DOC-EXAMPLE-BUCKET1-us-east-1` accede al bucket S3 `DOC-EXAMPLE-BUCKET1-logs-us-east-1` con prefisso `3-DOC-EXAMPLE-BUCKET1-us-east-1`
   - `1-DOC-EXAMPLE-BUCKET1-us-west-2` accede al bucket S3 `DOC-EXAMPLE-BUCKET1-logs-us-west-2` con prefisso `1-DOC-EXAMPLE-BUCKET1-us-west-2`
   - `2-DOC-EXAMPLE-BUCKET1-us-west-2` accede al bucket S3 `DOC-EXAMPLE-BUCKET1-logs-us-west-2` con prefisso `2-DOC-EXAMPLE-BUCKET1-us-west-2`

3. Per ogni bucket di registrazione di log di destinazione, concedi le autorizzazioni per la consegna di log di accesso al server utilizzando una ACL di bucket o una policy del bucket:

   - Aggiorna la policy del bucket (consigliato): per concedere autorizzazioni al principale del servizio di registrazione di log, utilizza il comando `put-bucket-policy` seguente. Sostituisci *`example-s3-destination-bucket-logs`* con il nome del tuo bucket di destinazione.

     ```
     aws s3api put-bucket-policy --bucket example-s3-destination-bucket-logs --policy
      file://policy.json
     ```

     `Policy.json` è un documento JSON nella cartella corrente che contiene la policy del bucket seguente. Per utilizzare questa policy del bucket, sostituisci *`user input placeholders`* con le tue informazioni. Nella policy seguente, *`example-s3-destination-bucket-logs`* è il bucket di destinazione in cui verranno distribuiti i log degli accessi al server e *`example-s3-source-bucket`* è il bucket di origine. *`SOURCE-ACCOUNT-ID`* è l' Account AWS proprietario del bucket di origine.

     ```
     {
         "Version": "2012-10-17",
         "Statement": [
             {
                 "Sid": "S3ServerAccessLogsPolicy",
                 "Effect": "Allow",
                 "Principal": {
                     "Service": "logging.s3.amazonaws.com"
     ```

```
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::example-s3-destination-bucket-logs/*",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::example-s3-source-bucket"
                },
                "StringEquals": {
                    "aws:SourceAccount": "SOURCE-ACCOUNT-ID"
                }
            }
        }
    ]
}
```

- Aggiorna l'ACL bucket: per concedere le autorizzazioni al gruppo di distribuzione dei log S3, utilizza il comando `put-bucket-acl` seguente. Sostituisci *example-s3-destination-bucket-logs* con il nome del tuo bucket di destinazione (target).

```
aws s3api put-bucket-acl --bucket example-s3-destination-bucket-logs  --grant-
write URI=http://acs.amazonaws.com/groups/s3/LogDelivery --grant-read-acp
 URI=http://acs.amazonaws.com/groups/s3/LogDelivery
```

4. Quindi, crea un file `logging.json` che contenga la configurazione di registrazione di log (in base a uno dei tre esempi che seguono). Dopo aver creato il file `logging.json`, puoi applicare la configurazione di registrazione di log utilizzando il comando `put-bucket-logging` seguente. Sostituisci *example-s3-destination-bucket-logs* con il nome del tuo bucket di destinazione (target).

```
aws s3api put-bucket-logging --bucket example-s3-destination-bucket-logs --bucket-
logging-status file://logging.json
```

> **Note**
>
> Invece di usare questo comando `put-bucket-logging` per applicare la configurazione di registrazione di log su ogni bucket di destinazione, puoi usare uno degli script bash forniti nel passaggio successivo. Per utilizzare questi script, devi creare i file `policy.json` e `logging.json`, come descritto in questa procedura.

Il file `logging.json` è un documento JSON nella cartella corrente che contiene la configurazione della registrazione di log. Se un bucket di destinazione utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto, la configurazione di registrazione di log non può contenere concessioni di destinazione (target). Per ulteriori informazioni, consulta Autorizzazioni per la distribuzione dei registri.

Example – **`logging.json`** senza concessioni relative alla destinazione (target)

Il seguente file di esempio `logging.json` contiene concessioni di destinazione (target). Pertanto, puoi applicare questa configurazione a un bucket di destinazione (target) che utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto.

```
{
    "LoggingEnabled": {
        "TargetBucket": "example-s3-destination-bucket-logs",
        "TargetPrefix": "example-s3-destination-bucket/"
     }
 }
```

Example – **`logging.json`** con concessioni relative alla destinazione (target)

Il seguente file di esempio `logging.json` contiene concessioni di destinazione (target).

Se un bucket di destinazione utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto, non sarà possibile includere concessioni di destinazione (target) nella configurazione PutBucketLogging. Per ulteriori informazioni, consulta Autorizzazioni per la distribuzione dei registri.

```
{
    "LoggingEnabled": {
        "TargetBucket": "example-s3-destination-bucket-logs",
        "TargetPrefix": "example-s3-destination-bucket/",
        "TargetGrants": [
             {
                "Grantee": {
                    "Type": "AmazonCustomerByEmail",
                    "EmailAddress": "user@example.com"
                 },
                "Permission": "FULL_CONTROL"
             }
         ]
     }
 }
```

Example – **`logging.json`** con il formato della chiave dell'oggetto di log impostato sull'ora dell'evento S3

Il file `logging.json` seguente modifica il formato della chiave dell'oggetto di log in Ora evento S3. Per informazioni sull'impostazione del formato della chiave dell'oggetto di log, consulta the section called “Come si abilita il recapito dei log?”.

```
{
  "LoggingEnabled": {
      "TargetBucket": "example-s3-destination-bucket-logs",
      "TargetPrefix": "example-s3-destination-bucket/",
      "TargetObjectKeyFormat": {
          "PartitionedPrefix": {
              "PartitionDateSource": "EventTime"
          }
       }
   }
}
```

5. Utilizza uno dei seguenti script bash per aggiungere la registrazione di log degli accessi per tutti i bucket nel tuo account. Sostituisci *example-s3-destination-bucket-logs* con il nome

del bucket di destinazione (target) e sostituisci *us-west-2* con il nome della regione in cui si trovano i bucket.

> **Note**
>
> Questo script funziona solo se tutti i bucket si trovano nella stessa regione. Se ci sono bucket in più regioni, è necessario modificare lo script.

Example – Concedi l'accesso con le policy del bucket e aggiungi la registrazione per i bucket nel tuo account

```
loggingBucket='example-s3-destination-bucket-logs'
region='us-west-2'


# Create the logging bucket.
aws s3 mb s3://$loggingBucket --region $region

aws s3api put-bucket-policy --bucket $loggingBucket --policy file://policy.json

# List the buckets in this account.
buckets="$(aws s3 ls | awk '{print $3}')"

# Put a bucket logging configuration on each bucket.
for bucket in $buckets
    do
      # This if statement excludes the logging bucket.
      if [ "$bucket" != "$loggingBucket" ] ; then
          continue;
      fi
      printf '{
        "LoggingEnabled": {
          "TargetBucket": "%s",
          "TargetPrefix": "%s/"
      }
    }' "$loggingBucket" "$bucket"  > logging.json
    aws s3api put-bucket-logging --bucket $bucket --bucket-logging-status file://
logging.json
    echo "$bucket done"
done
```

```
  rm logging.json

  echo "Complete"
```

Example – Concedi l'accesso con ACL di bucket e aggiungi la registrazione per i bucket nel tuo account

```
  loggingBucket='example-s3-destination-bucket-logs'
  region='us-west-2'


  # Create the logging bucket.
  aws s3 mb s3://$loggingBucket --region $region

  aws s3api put-bucket-acl --bucket $loggingBucket --grant-write URI=http://
acs.amazonaws.com/groups/s3/LogDelivery --grant-read-acp URI=http://
acs.amazonaws.com/groups/s3/LogDelivery

  # List the buckets in this account.
  buckets="$(aws s3 ls | awk '{print $3}')"

  # Put a bucket logging configuration on each bucket.
  for bucket in $buckets
      do
        # This if statement excludes the logging bucket.
        if [ "$bucket" != "$loggingBucket" ] ; then
            continue;
        fi
        printf '{
          "LoggingEnabled": {
            "TargetBucket": "%s",
            "TargetPrefix": "%s/"
        }
      }' "$loggingBucket" "$bucket"  > logging.json
      aws s3api put-bucket-logging --bucket $bucket --bucket-logging-status file://
logging.json
      echo "$bucket done"
  done

  rm logging.json
```

```
  echo "Complete"
```

Verifica della configurazione dei log degli accessi al server

Dopo aver abilitato la registrazione degli accessi al server, completa la procedura riportata di seguito:

- Accedi al bucket di destinazione e verifica che i file di log vengano distribuiti. Dopo che i log di accesso sono stati configurati, potrebbe essere necessaria più di un'ora prima che tutte le richieste vengano registrate e distribuite correttamente. Puoi anche verificare automaticamente la consegna dei log utilizzando i parametri delle richieste di Amazon S3 e configurando gli CloudWatch allarmi Amazon per questi parametri. Per ulteriori informazioni, consulta Monitoraggio delle metriche con Amazon CloudWatch.
- Verifica di essere in grado di aprire e leggere il contenuto dei file di log.

Per informazioni sulla risoluzione dei problemi relativi alla registrazione degli accessi al server, consultare Risoluzione dei problemi di registrazione degli accessi al server.

Formato del log di accesso al server Amazon S3

La registrazione degli accessi al server fornisce record dettagliati per le richieste che sono effettuate a un bucket Amazon S3. È possibile utilizzare i log degli accessi al server per i seguenti scopi:

- Esecuzione di controlli di sicurezza e degli accessi
- Informazioni sulla base di clienti
- Informazioni sulla fatturazione Amazon S3

In questa sezione viene descritto il formato e altri dettagli sui file di log di accesso al server Amazon S3.

I file dei log di accesso al server sono composti da una sequenza di record dei log delimitati da una nuova riga. Ogni record di log rappresenta una richiesta ed è composto da campi delimitati da spazio.

Di seguito è riportato un esempio di log composto da cinque record di log.

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DOC-EXAMPLE-BUCKET1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 3E57427F3EXAMPLE
 REST.GET.VERSIONING - "GET /DOC-EXAMPLE-BUCKET1?versioning HTTP/1.1" 200 - 113 - 7 -
```

```
 "-" "S3Console/0.4" - s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/
XV/VLi31234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader DOC-EXAMPLE-BUCKET1.s3.us-
west-1.amazonaws.com TLSV1.2 arn:aws:s3:us-west-1:123456789012:accesspoint/example-AP
 Yes
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DOC-EXAMPLE-BUCKET1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 891CE47D2EXAMPLE
 REST.GET.LOGGING_STATUS - "GET /DOC-EXAMPLE-BUCKET1?logging HTTP/1.1" 200 -
 242 - 11 - "-" "S3Console/0.4" - 9vKBE6vMhrNiWHZmb2L0mXOcqPGzQOI5XLnCtZNPxev+Hf
+7tpT6sxDwDty4LHBUOZJG96N1234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader DOC-
EXAMPLE-BUCKET1.s3.us-west-1.amazonaws.com TLSV1.2 - -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DOC-EXAMPLE-BUCKET1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be A1206F460EXAMPLE
 REST.GET.BUCKETPOLICY - "GET /DOC-EXAMPLE-BUCKET1?policy HTTP/1.1" 404
 NoSuchBucketPolicy 297 - 38 - "-" "S3Console/0.4" - BNaBsXZQQDbssi6xMBdBU2sLt
+Yf5kZDmeBUP35sFoKa3sLLeMC78iwEIWxs99CRUrbS4n11234= SigV4 ECDHE-RSA-AES128-GCM-SHA256
 AuthHeader DOC-EXAMPLE-BUCKET1.s3.us-west-1.amazonaws.com TLSV1.2 - Yes
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DOC-EXAMPLE-BUCKET1 [06/Feb/2019:00:01:00 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 7B4A0FABBEXAMPLE
 REST.GET.VERSIONING - "GET /DOC-EXAMPLE-BUCKET1?versioning HTTP/1.1" 200 -
 113 - 33 - "-" "S3Console/0.4" - Ke1bUcazaN1jWuUlPJaxF64cQVpUEhoZKEG/hmy/gijN/
I1DeWqDfFvnpybfEseEME/u7ME1234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader DOC-
EXAMPLE-BUCKET1.s3.us-west-1.amazonaws.com TLSV1.2 - -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DOC-EXAMPLE-BUCKET1 [06/Feb/2019:00:01:57 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DD6CC733AEXAMPLE REST.PUT.OBJECT s3-dg.pdf "PUT /DOC-EXAMPLE-BUCKET1/
s3-dg.pdf HTTP/1.1" 200 - - 4406583 41754 28 "-" "S3Console/0.4" -
 10S62Zv81kBW7BB6SX4XJ48o6kpcl6LPwEoizZQQxJd5qDSCTLX0TgS37kYUBKQW3+bPdrg1234= SigV4
 ECDHE-RSA-AES128-SHA AuthHeader DOC-EXAMPLE-BUCKET1.s3.us-west-1.amazonaws.com TLSV1.2
 - Yes
```

> **Note**
>
> Qualsiasi campo può essere impostato su - per indicare che i dati sono sconosciuti o non disponibili, oppure che il campo non è applicabile a questa richiesta.

Argomenti

- Campi dei record dei log

- Registrazione aggiuntiva per operazioni di copia
- Informazioni sui log di accesso personalizzati
- Considerazioni in materia di programmazione per il formato esteso dei log di accesso al server

Campi dei record dei log

L'elenco di seguito descrive i campi dei record di log.

Proprietario del bucket

L'ID utente canonico del proprietario del bucket di origine. L'ID utente canonico è un'altra forma di ID. Account AWS Per ulteriori informazioni sull'ID utente canonico, consultare la sezione relativa agli identificatori Account AWS nella Riferimenti generali di AWS. Per informazioni su come trovare l'ID utente canonico per il tuo account, consulta Ricerca dell'ID utente canonico per l' Account AWS.

Esempio di inserimento

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Bucket

Il nome del bucket riguardo al quale è stata elaborata la richiesta. Se il sistema riceve una richiesta non corretta e non riesce a determinare il bucket, tale richiesta non apparirà in alcun log di accesso al server.

Esempio di inserimento

```
DOC-EXAMPLE-BUCKET1
```

Orario

L'ora di ricezione della richiesta; queste date e ore sono in formato UTC. Il formato, utilizzando la terminologia `strftime()`, è il seguente: `[%d/%b/%Y:%H:%M:%S %z]`

Esempio di inserimento

```
[06/Feb/2019:00:00:38 +0000]
```

IP remoto

Indirizzo IP apparente del richiedente. Dei proxy e firewall intermedi potrebbero oscurare l'indirizzo IP effettivo della macchina che effettua la richiesta.

Esempio di inserimento

```
192.0.2.3
```

Richiedente

L'ID utente canonico del richiedente o - per richieste non autenticate. Se il richiedente era un utente IAM, questo campo restituisce il nome utente IAM del richiedente insieme a Utente root dell'account AWS quello a cui appartiene l'utente IAM. Questo identificatore è lo stesso che viene usato per accedere a scopi di controllo.

Esempio di inserimento

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Se il richiedente utilizza un ruolo presunto, questo campo restituisce il ruolo IAM presunto.

Esempio di inserimento

```
arn:aws:sts::123456789012:assumed-role/roleName/test-role
```

ID di richiesta

Una stringa generata da Amazon S3 per identificare in maniera univoca ogni richiesta.

Esempio di inserimento

```
3E57427F33A59F07
```

Operazione

L'operazione elencata qui viene dichiarata come `SOAP.`*`operation`*, `REST.`*`HTTP_method`*`.`*`resource_type`*, `WEBSITE.`*`HTTP_method`*`.`*`resource_type`*, oppure `BATCH.DELETE.OBJECT`, oppure `S3.action.resource_type` per Ciclo di vita e registrazione.

Esempio di inserimento

```
REST.PUT.OBJECT
```

Chiave

La parte chiave (nome dell'oggetto) della richiesta.

Esempio di inserimento

```
/photos/2019/08/puppy.jpg
```

Request-URI

La parte `Request-URI` del messaggio di richiesta HTTP.

Esempio di inserimento

```
"GET /DOC-EXAMPLE-BUCKET1/photos/2019/08/puppy.jpg?x-foo=bar HTTP/1.1"
```

Stato HTTP

Il codice di stato HTTP numerico della risposta.

Esempio di inserimento

```
200
```

Codice di errore

Amazon S3 Codice di errore o `-` se non si è verificato alcun errore.

Esempio di inserimento

```
NoSuchBucket
```

Byte inviati

Il numero di byte della risposta inviati, a esclusione di overhead di protocollo HTTP, o `-` se uguale a zero.

Esempio di inserimento

```
2662992
```

Dimensione oggetto

La dimensione totale dell'oggetto in questione.

Esempio di inserimento

```
3462992
```

Tempo totale

Il numero di millisecondi durante i quali la richiesta è stata in transito dalla prospettiva del server. Questo valore viene misurato dal momento in cui si riceve la richiesta al momento in cui viene inviato l'ultimo byte di risposta. Le misurazioni effettuate dalla prospettiva del cliente potrebbero essere più lunghe in ragione della latenza di rete.

Esempio di inserimento

```
70
```

Tempo di rotazione

Il numero di millisecondi che sono stati necessari ad Amazon S3 per elaborare la richiesta. Questo valore viene misurato dal momento in cui si riceve l'ultimo byte della richiesta al momento in cui viene inviato il primo byte di risposta.

Esempio di inserimento

```
10
```

Referer

Il valore dell'intestazione HTTP `Referer`, se presente. Gli utenti-agenti HTTP (ad esempio, i browser) generalmente impostano questa intestazione sull'URL della pagina di collegamento o incorporazione quando viene effettuata una richiesta.

Esempio di inserimento

```
"http://www.example.com/webservices"
```

User-Agent

Il valore dell'intestazione HTTP `User-Agent`.

Esempio di inserimento

```
"curl/7.15.1"
```

Versione ID

L'ID della versione nella richiesta oppure `-` se l'operazione non prevede un parametro `versionId`.

Esempio di inserimento

```
3HL4kqtJvjVBH40Nrjfkd
```

ID host

`x-amz-id-2` o ID richiesta esteso Amazon S3.

Esempio di inserimento

```
s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Signature Version

La versione della firma, `SigV2` o `SigV4`, utilizzata per autenticare la richiesta o un `-` per richieste non autenticate.

Esempio di inserimento

```
SigV2
```

Pacchetti di crittografia

La crittografia Secure Sockets Layer (SSL) negoziata per richieste HTTPS o `-` per HTTP.

Esempio di inserimento

```
ECDHE-RSA-AES128-GCM-SHA256
```

Tipo di autenticazione

Il tipo di autenticazione della richiesta utilizzato: `AuthHeader` per intestazioni autenticate, `QueryString` per stringa di query (URL prefirmato) o `-` per richieste non autenticate.

Esempio di inserimento

```
AuthHeader
```

Intestazione dell'host

L'endpoint utilizzato per connettersi ad Amazon S3.

Esempio di inserimento

```
s3.us-west-2.amazonaws.com
```

Alcune regioni meno recenti supportano gli endpoint legacy. Potresti vedere questi endpoint nei log o nei log di accesso al server. AWS CloudTrail Per ulteriori informazioni, consulta Endpoint legacy. Per un elenco completo degli endpoint e delle regioni Amazon S3, consultare la sezione relativa a endpoint e quote di Amazon S3 nella Riferimenti generali di Amazon Web Services.

Versione TLS

La versione di Transport Layer Security (TLS) negoziata dal client. Il valore è uno dei seguenti: `TLSv1.1`, `TLSv1.2`, `TLSv1.3` o `-` se non è stato utilizzato TLS.

Esempio di inserimento

```
TLSv1.2
```

Nome della risorsa Amazon (ARN) del punto di accesso

L'Amazon Resource Name (ARN) del punto di accesso della richiesta. Se il nome della risorsa Amazon (ARN) del punto di accesso ha un formato non valido oppure non viene utilizzato, il campo conterrà `-`. Per ulteriori informazioni sui punti di accesso, consulta la sezione Utilizzo degli access point. Per maggiori informazioni sui nomi delle risorse Amazon (ARN), consulta Nome della risorsa Amazon (ARN) in Riferimenti generali di AWS .

Esempio di inserimento

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

aclRequired

Una stringa che indica se la richiesta richiede una lista di controllo degli accessi (ACL) per l'autorizzazione. Se la richiesta richiede una ACL per l'autorizzazione, la stringa è `Yes`. Se non sono richieste ACL, la stringa è `-`. Per ulteriori informazioni sulle ACL, consulta Panoramica delle liste di controllo accessi (ACL). Per ulteriori informazioni sull'utilizzo del campo `aclRequired` per disabilitare le ACL, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

Esempio di inserimento

```
Yes
```

Registrazione aggiuntiva per operazioni di copia

Un'operazione di copia implica un `GET` e un `PUT`. Per questa ragione, vengono registrati due report quando si effettua un'operazione di logging. La tabella precedente descrive i campi che si riferiscono alla parte `PUT` dell'operazione. L'elenco di seguito descrive i campi nel record che si riferiscono alla parte `GET` dell'operazione di copia.

Proprietario del bucket

L'ID utente canonico del bucket archivia l'oggetto che viene copiato. L'ID utente canonico è un'altra forma di ID. Account AWS Per ulteriori informazioni sull'ID utente canonico, consultare la sezione relativa agli identificatori Account AWS nella Riferimenti generali di AWS. Per informazioni su come trovare l'ID utente canonico per il tuo account, consulta Ricerca dell'ID utente canonico per l' Account AWS.

Esempio di inserimento

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Bucket

Nome del bucket che archivia l'oggetto che viene copiato.

Esempio di inserimento

```
DOC-EXAMPLE-BUCKET1
```

Orario

L'ora di ricezione della richiesta; queste date e ore sono in formato UTC. Il formato, utilizzando la terminologia `strftime()`, è il seguente: `[%d/%B/%Y:%H:%M:%S %z]`

Esempio di inserimento

```
[06/Feb/2019:00:00:38 +0000]
```

IP remoto

Indirizzo IP apparente del richiedente. Dei proxy e firewall intermedi potrebbero oscurare l'indirizzo IP effettivo della macchina che effettua la richiesta.

Esempio di inserimento

```
192.0.2.3
```

Richiedente

L'ID utente canonico del richiedente o `-` per richieste non autenticate. Se il richiedente era un utente IAM, questo campo restituirà il nome utente IAM del richiedente insieme a Utente root dell'account AWS quello a cui appartiene l'utente IAM. Questo identificatore è lo stesso che viene usato per accedere a scopi di controllo.

Esempio di inserimento

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Se il richiedente utilizza un ruolo presunto, questo campo restituisce il ruolo IAM presunto.

Esempio di inserimento

```
arn:aws:sts::123456789012:assumed-role/roleName/test-role
```

ID di richiesta

Una stringa generata da Amazon S3 per identificare in maniera univoca ogni richiesta.

Esempio di inserimento

```
3E57427F33A59F07
```

Operazioni

Le operazioni qui elencate vengono dichiarate come `SOAP.operation`, `REST.HTTP_method.resource_type`, `WEBSITE.HTTP_method.resource_type` oppure `BATCH.DELETE.OBJECT`.

Esempio di inserimento

```
REST.COPY.OBJECT_GET
```

Chiave

La "chiave" (nome oggetto) dell'oggetto che viene copiato o `-` se l'operazione non prevede un parametro chiave.

Esempio di inserimento

```
/photos/2019/08/puppy.jpg
```

Request-URI

La parte `Request-URI` del messaggio di richiesta HTTP.

Esempio di inserimento

```
"GET /DOC-EXAMPLE-BUCKET1/photos/2019/08/puppy.jpg?x-foo=bar"
```

Stato HTTP

Il codice di stato HTTP numerico della porzione `GET` dell'operazione di copia.

Esempio di inserimento

```
200
```

Codice di errore

Codice di errore Amazon S3 della porzione `GET` dell'operazione di copia o `-` se non si è verificato alcun errore.

Esempio di inserimento

```
NoSuchBucket
```

Byte inviati

Numero di byte della risposta inviati, a esclusione di overhead di protocollo HTTP, o - se uguale a zero.

Esempio di inserimento

```
2662992
```

Dimensione oggetto

La dimensione totale dell'oggetto in questione.

Esempio di inserimento

```
3462992
```

Tempo totale

Il numero di millisecondi durante i quali la richiesta è stata in transito dalla prospettiva del server. Questo valore viene misurato dal momento in cui si riceve la richiesta al momento in cui viene inviato l'ultimo byte di risposta. Le misurazioni effettuate dalla prospettiva del cliente potrebbero essere più lunghe in ragione della latenza di rete.

Esempio di inserimento

```
70
```

Tempo di rotazione

Il numero di millisecondi che sono stati necessari ad Amazon S3 per elaborare la richiesta. Questo valore viene misurato dal momento in cui si riceve l'ultimo byte della richiesta al momento in cui viene inviato il primo byte di risposta.

Esempio di inserimento

```
10
```

Referer

Il valore dell'intestazione HTTP `Referer`, se presente. Gli utenti-agenti HTTP (ad esempio, i browser) generalmente impostano questa intestazione sull'URL della pagina di collegamento o incorporazione quando viene effettuata una richiesta.

Esempio di inserimento

```
"http://www.example.com/webservices"
```

User-Agent

Il valore dell'intestazione HTTP `User-Agent`.

Esempio di inserimento

```
"curl/7.15.1"
```

Versione ID

ID versione dell'oggetto che viene copiato oppure `-` se l'intestazione `x-amz-copy-source` non specificava un parametro `versionId` come parte dell'origine della copia.

Esempio di inserimento

```
3HL4kqtJvjVBH40Nrjfkd
```

ID host

`x-amz-id-2` o ID richiesta esteso Amazon S3.

Esempio di inserimento

```
s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Signature Version

Versione della firma, `SigV2` o `SigV4`, utilizzata per autenticare la richiesta o `-` per richieste non autenticate.

Esempio di inserimento

```
SigV4
```

Pacchetti di crittografia

La crittografia Secure Sockets Layer (SSL) negoziata per richieste HTTPS o `-` per HTTP.

Esempio di inserimento

```
ECDHE-RSA-AES128-GCM-SHA256
```

Tipo di autenticazione

Tipo di autenticazione della richiesta utilizzato: `AuthHeader` per intestazioni autenticate, `QueryString` per stringa di query (URL prefirmati) o `-` per richieste non autenticate.

Esempio di inserimento

```
AuthHeader
```

Intestazione dell'host

L'endpoint utilizzato per connettersi ad Amazon S3.

Esempio di inserimento

```
s3.us-west-2.amazonaws.com
```

Alcune regioni meno recenti supportano gli endpoint legacy. Potresti vedere questi endpoint nei log o nei log di accesso al server. AWS CloudTrail Per ulteriori informazioni, consulta Endpoint legacy. Per un elenco completo degli endpoint e delle regioni Amazon S3, consultare la sezione relativa a endpoint e quote di Amazon S3 nella Riferimenti generali di Amazon Web Services.

Versione TLS

La versione di Transport Layer Security (TLS) negoziata dal client. Il valore è uno dei seguenti: `TLSv1.1`, `TLSv1.2`, `TLSv1.3` o `-` se non è stato utilizzato TLS.

Esempio di inserimento

```
TLSv1.2
```

Nome della risorsa Amazon (ARN) del punto di accesso

L'Amazon Resource Name (ARN) del punto di accesso della richiesta. Se il nome della risorsa Amazon (ARN) del punto di accesso ha un formato non valido oppure non viene utilizzato, il campo conterrà `-`. Per ulteriori informazioni sui punti di accesso, consulta la sezione Utilizzo degli access point. Per maggiori informazioni sui nomi delle risorse Amazon (ARN), consulta Nome della risorsa Amazon (ARN) in Riferimenti generali di AWS .

Esempio di inserimento

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

aclRequired

Una stringa che indica se la richiesta richiede una lista di controllo degli accessi (ACL) per l'autorizzazione. Se la richiesta richiede una ACL per l'autorizzazione, la stringa è `Yes`. Se non sono richieste ACL, la stringa è `-`. Per ulteriori informazioni sulle ACL, consulta Panoramica delle liste di controllo accessi (ACL). Per ulteriori informazioni sull'utilizzo del campo `aclRequired` per disabilitare le ACL, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

Esempio di inserimento

```
Yes
```

Informazioni sui log di accesso personalizzati

È possibile includere informazioni personalizzate da memorizzare nel record del log di accesso per una richiesta. A tale scopo, aggiungere un parametro di stringa query personalizzato all'URL per la richiesta. Amazon S3 ignora i parametri di stringa di query che iniziano con `x-`, ma include quelli nel record del log degli accessi per la richiesta, come parte del campo `Request-URI` del record del log.

Ad esempio, una richiesta `GET` per `"s3.amazonaws.com/DOC-EXAMPLE-BUCKET1/photos/2019/08/puppy.jpg?x-user=johndoe"` funziona come la richiesta `"s3.amazonaws.com/DOC-EXAMPLE-BUCKET1/photos/2019/08/puppy.jpg"`, ad eccezione del fatto che la stringa `"x-user=johndoe"` è inclusa nel campo `Request-URI` per il record di log associato. Questa funzionalità è disponibile solo nell'interfaccia REST.

Considerazioni in materia di programmazione per il formato esteso dei log di accesso al server

Occasionalmente è possibile estendere il formato del report del log degli accessi aggiungendo nuovi campi alla fine di ogni linea. Pertanto, è necessario che il codice che analizza i log degli accessi al server sia in grado di gestire i campi che potrebbero non essere riconosciuti.

Eliminazione dei file di log Amazon S3

Un bucket Amazon S3 con la registrazione degli accessi al server abilitata può accumulare nel tempo molti oggetti del log del server. L'applicazione potrebbe necessitare di questi log di accesso per un periodo specifico dopo la creazione e successivamente potrebbe eliminarli. Puoi utilizzare la configurazione del ciclo di vita in Amazon S3 per impostare regole in modo che Amazon S3 accodi automaticamente questi oggetti per l'eliminazione alla fine del loro ciclo di vita.

È possibile definire una configurazione del ciclo di vita per un sottoinsieme di oggetti nel bucket S3 utilizzando un prefisso condiviso. Se è stato specificato un prefisso nella configurazione della registrazione degli accessi al server, è possibile impostare una regola di configurazione del ciclo di vita per eliminare gli oggetti del log con quel prefisso.

Supponi, ad esempio, che i tuoi oggetti di log abbiano il prefisso `logs/`. Puoi impostare una regola di configurazione del ciclo di vita per eliminare tutti gli oggetti nel bucket che hanno il prefisso `logs/` dopo un periodo di tempo specificato.

Per ulteriori informazioni sulla configurazione del ciclo di vita, consulta Gestione del ciclo di vita dello storage.

Per ulteriori informazioni sulla registrazione di accesso al server, consulta Registrazione delle richieste con registrazione dell'accesso al server.

Utilizzo dei log degli accessi al server Amazon S3 per identificare le richieste

Puoi identificare le richieste Amazon S3 con i log degli accessi al server Amazon S3.

> **Note**
>
> - Per identificare le richieste Amazon S3, ti consigliamo di utilizzare eventi AWS CloudTrail relativi ai dati anziché i log di accesso al server Amazon S3. CloudTrail gli eventi relativi ai

dati sono più facili da configurare e contengono più informazioni. Per ulteriori informazioni, consulta Identificazione delle richieste Amazon S3 tramite CloudTrail.

- A seconda del numero di richieste di accesso ricevute, l'analisi dei log potrebbe richiedere più risorse o tempo rispetto all'utilizzo degli eventi relativi ai CloudTrail dati.

Argomenti

Esecuzione di query sui log degli accessi per le richieste tramite Amazon Athena

Puoi identificare le richieste ad Amazon S3 con i log degli accessi ad Amazon S3 utilizzando Amazon Athena.

Amazon S3 archivia i log degli accessi al server come oggetti in un bucket S3. Spesso è più facile utilizzare uno strumento in grado di analizzare i log in Amazon S3. Athena supporta l'analisi di oggetti S3 e può essere utilizzato per eseguire query sui log degli accessi Amazon S3.

Example

L'esempio seguente mostra come eseguire query sui log degli accessi al server Amazon S3 in Amazon Athena. Sostituisci *`user input placeholders`* che trovi nei seguenti esempi con le tue informazioni.

> **Note**
>
> Per specificare una posizione Amazon S3 in una query Athena, devi fornire un URI S3 per il bucket in cui vengono distribuiti i log. Questo URI deve includere il nome e il prefisso del bucket nel seguente formato: `s3://`*`example-s3-bucket1`*`-logs/`*`prefix`*`/`

1. Aprire la console Athena all'indirizzo https://console.aws.amazon.com/athena/.
2. Nel Query Editor esegui un comando simile al seguente. Sostituisci *`s3_access_logs_db`* con il nome che desideri assegnare al database.

```
CREATE DATABASE s3_access_logs_db
```

> **Note**
>
> È consigliabile creare il database nello stesso Regione AWS bucket S3.

3. Nel Query Editor eseguire un comando simile al seguente per creare uno schema di tabella nel database creato nella fase 2. Sostituisci *s3_access_logs_db.mybucket_logs* con il nome che desideri assegnare alla tabella. I valori dei tipi di dati `STRING` e `BIGINT` sono le proprietà del log di accesso. È possibile eseguire query su queste proprietà in Athena. Per `LOCATION`, immettere il percorso del prefisso e il bucket S3 come indicato in precedenza.

```
CREATE EXTERNAL TABLE `s3_access_logs_db.mybucket_logs`(
  `bucketowner` STRING,
  `bucket_name` STRING,
  `requestdatetime` STRING,
  `remoteip` STRING,
  `requester` STRING,
  `requestid` STRING,
  `operation` STRING,
  `key` STRING,
  `request_uri` STRING,
  `httpstatus` STRING,
  `errorcode` STRING,
  `bytessent` BIGINT,
  `objectsize` BIGINT,
  `totaltime` STRING,
  `turnaroundtime` STRING,
  `referrer` STRING,
  `useragent` STRING,
  `versionid` STRING,
  `hostid` STRING,
  `sigv` STRING,
  `ciphersuite` STRING,
  `authtype` STRING,
  `endpoint` STRING,
  `tlsversion` STRING,
  `accesspointarn` STRING,
  `aclrequired` STRING)
ROW FORMAT SERDE
```

```
  'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'input.regex'='([^ ]*) ([^ ]*) \\[(.*?)\\] ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 ([^ ]*) (\"[^\"]*\"|-) (-|[0-9]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 (\"[^\"]*\"|-) ([^ ]*)(?: ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 ([^ ]*))?.*$')
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET1-logs/prefix/'
```

4. Nel riquadro di navigazione, in Database, scegliere il database.
5. In Tables (Tabelle), scegliere Preview table (Anteprima tabella) accanto al nome della tabella.

   Nel pannello Results (Risultati), dovrebbero essere visualizzati i dati dai log di accesso al server, come `bucketowner`, `bucket`, `requestdatetime` e così via. Questo indica che la tabella Athena è stata creata correttamente. È ora possibile eseguire query sui log degli accessi al server Amazon S3.

Example - Visualizza chi ha eliminato un oggetto e quando (timestamp, indirizzo IP e utente IAM)

```
SELECT requestdatetime, remoteip, requester, key
FROM s3_access_logs_db.mybucket_logs
WHERE key = 'images/picture.jpg' AND operation like '%DELETE%';
```

Example - Visualizza tutte le operazioni eseguite da un utente IAM

```
SELECT *
FROM s3_access_logs_db.mybucket_logs
WHERE requester='arn:aws:iam::123456789123:user/user_name';
```

Example - Visualizza tutte le operazioni eseguite su un oggetto in un periodo di tempo specifico

```
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 : visualizza la quantità di dati trasferiti da un indirizzo IP specifico in un determinato periodo di tempo

```
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**
>
> Per ridurre il periodo di conservazione dei log, puoi creare una configurazione del ciclo di vita S3 per il bucket dei log degli accessi al server. Crea regole di configurazione del ciclo di vita per rimuovere i file di log periodicamente. In questo modo, si riduce la quantità di dati analizzati da Athena per ogni query. Per ulteriori informazioni, consulta Impostazione di una configurazione del ciclo di vita su un bucket.

Identificazione delle richieste di Signature versione 2 tramite i log degli accessi ad Amazon S3

Il supporto di Amazon S3 per Signature Version 2 sta per essere disattivato in quanto obsoleto. Dopo, Amazon S3 non accetterà più le richieste che usano Signature Version 2 e tutte le richieste dovranno usare la firma Signature Version 4. Puoi identificare le richieste di accesso a Signature versione 2 utilizzando i log degli accessi ad Amazon S3.

> **Note**
>
> Per identificare le richieste Signature versione 2, ti consigliamo di utilizzare eventi AWS CloudTrail relativi ai dati anziché i log di accesso al server Amazon S3. CloudTrail gli eventi di

dati sono più facili da configurare e contengono più informazioni rispetto ai log di accesso al server. Per ulteriori informazioni, consulta Identificazione delle richieste Amazon S3 Signature versione 2 mediante CloudTrail.

Example - Visualizza tutti i richiedenti che inviano traffico Signature versione 2

```
SELECT requester, sigv, Count(sigv) as sigcount
FROM s3_access_logs_db.mybucket_logs
GROUP BY requester, sigv;
```

Identificazione delle richieste di accesso agli oggetti tramite i log degli accessi Amazon S3

Puoi utilizzare query sui log degli accessi al server Amazon S3 per identificare le richieste di accesso a oggetti Amazon S3, per operazioni come `GET`, `PUT` e `DELETE`, e ottenere ulteriori informazioni su queste richieste.

L'esempio di query Amazon Athena seguente mostra come ottenere tutte le richieste di oggetti `PUT` per Amazon S3 da un log degli accessi al server.

Example : visualizza tutti i richiedenti che inviano richieste **`PUT`** di oggetti in un determinato periodo

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
FROM s3_access_logs_db
WHERE operation='REST.PUT.OBJECT' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

L'esempio di query Amazon Athena seguente mostra come ottenere tutte le richieste `GET` di oggetti per Amazon S3 dal log degli accessi al server.

Example : visualizza tutti i richiedenti che inviano richieste **`GET`** di oggetti in un determinato periodo

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
```

```
FROM s3_access_logs_db
WHERE operation='REST.GET.OBJECT' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

L'esempio di query Amazon Athena seguente mostra come ottenere tutte le richieste anonime ai bucket S3 dal log degli accessi al server.

Example : visualizza tutti i richiedenti anonimi che effettuano richieste a un bucket in un determinato periodo

```
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 seguente query Amazon Athena mostra come identificare tutte le richieste ai bucket S3 che richiedono una lista di controllo degli accessi (ACL) per l'autorizzazione. È possibile utilizzare queste informazioni per eseguire la migrazione di tali autorizzazioni ACL nelle policy dei bucket appropriate e disabilitare le ACL. Dopo aver creato queste policy di bucket, puoi disabilitare le ACL per questi bucket. Per ulteriori informazioni sulla disabilitazione delle ACL, consulta Prerequisiti per la disabilitazione delle ACL.

Example : identifica tutte le richieste che richiedono una ACL per l'autorizzazione

```
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

- È possibile modificare l'intervallo di data in base alle esigenze.
- Questi esempi di query possono essere utili anche per il monitoraggio della sicurezza. Puoi rivedere i risultati per le chiamate `PutObject` o `GetObject` da indirizzi IP o richiedenti imprevisti o non autorizzati e per l'identificazione di eventuali richieste anonime ai bucket.
- La query recupera solo le informazioni a partire dall'orario in cui è stata abilitata la registrazione.
- Se si utilizzano i AWS CloudTrail log, vedere. Identificazione dell'accesso agli oggetti S3 utilizzando CloudTrail

Monitoraggio delle metriche con Amazon CloudWatch

CloudWatch I parametri di Amazon per Amazon S3 possono aiutarti a comprendere e migliorare le prestazioni delle applicazioni che utilizzano Amazon S3. Esistono diversi modi per utilizzarlo CloudWatch con Amazon S3.

Daily storage metrics for buckets (Parametri di archiviazione giornalieri per i bucket)

Monitora lo storage dei bucket utilizzando CloudWatch, che raccoglie ed elabora i dati di storage da Amazon S3 in parametri giornalieri leggibili. Questi parametri di storage per Amazon S3 vengono indicati una volta al giorno e sono disponibili per tutti i clienti senza costi aggiuntivi.

Request metrics (Parametri di richiesta)

Puoi monitorare le richieste di Amazon S3 per identificare rapidamente i problemi operativi e intraprendere le operazioni appropriate. I parametri sono disponibili a intervalli di 1 minuto dopo una determinata latenza di elaborazione. Questi CloudWatch parametri vengono fatturati alla stessa tariffa dei parametri CloudWatch personalizzati di Amazon. Per informazioni sui CloudWatch prezzi, consulta i CloudWatch prezzi di Amazon. Per informazioni su come ottenere questi parametri, consulta CloudWatch configurazioni delle metriche.

Quando sono abilitati, i parametri della richiesta vengono indicati per tutte le operazioni sull'oggetto. Per default, questi parametri da 1 minuto sono disponibili a livello di bucket di Amazon S3. Puoi anche definire un filtro per i parametri usando un prefisso condiviso, un tag oggetto o un punto di accesso:

- Punto di accesso: i punti di accesso sono endpoint di rete denominati e allegati a bucket che semplificano la gestione degli accessi ai dati su vasta scala per set di dati condivisi in S3. Con il filtro del punto di accesso, puoi ottenere informazioni dettagliate sull'utilizzo del punto di accesso. Per ulteriori informazioni sui punti di accesso, consulta la sezione Monitoraggio e registrazione degli access point.
- Prefisso: sebbene il modello di dati Amazon S3 abbia una struttura orizzontale, puoi applicare una gerarchia utilizzando i prefissi. Un prefisso è simile a un nome di directory che consente di raggruppare oggetti simili in un bucket. La console S3 supporta i prefissi con il concetto delle cartelle. Se si applica un filtro basato sul prefisso, gli oggetti con lo stesso prefisso verranno inclusi nella configurazione del parametro. Per ulteriori informazioni sui prefissi, consulta Organizzazione degli oggetti utilizzando i prefissi.
- Tag: puoi aggiungere tag, che sono coppie di nomi chiave-valore, agli oggetti. I tag consentono di trovare e organizzare gli oggetti in modo semplice Puoi inoltre possibile utilizzare i tag come filtro per le configurazioni dei parametri, in modo che nella configurazione vengano inclusi solo gli oggetti con tali tag. Per ulteriori informazioni sui tag degli oggetti, consulta Suddivisione in categorie dello storage utilizzando i tag.

Per allineare questi parametri a specifiche applicazioni business, flussi di lavoro o organizzazioni interne, puoi applicare un filtro in base a un prefisso condiviso, un tag oggetto o un punto di accesso.

Replication metrics (Parametri di replica)

Parametri di replica: monitorano il numero totale di operazioni API S3 in attesa di replica, la dimensione totale degli oggetti in attesa di replica e il tempo massimo di replica nella Regione AWS di destinazione e il numero totale di operazioni che non sono state replicate. Le regole di replica con il controllo del tempo di replica di S3 (S3 RTC) o le metriche di replica S3 abilitati pubblicheranno le metriche di replica.

Per ulteriori informazioni, consulta Monitoraggio dell'avanzamento con le metriche di replica e le notifiche eventi di Amazon S3 o Rispetto dei requisiti di conformità utilizzando S3 Replication Time Control (S3 RTC).

Parametri di Amazon S3 Storage Lens

Puoi pubblicare i parametri di utilizzo e attività di S3 Storage Lens su Amazon CloudWatch per creare una visione unificata dello stato di salute operativo nei dashboard. CloudWatch I parametri di S3 Storage Lens sono disponibili nello spazio dei nomi `AWS/S3/Storage-Lens`. L'opzione di CloudWatch pubblicazione è disponibile per i dashboard di S3 Storage Lens aggiornati a metriche

e consigli avanzati. Puoi abilitare l'opzione di CloudWatch pubblicazione per una configurazione del dashboard nuova o esistente in S3 Storage Lens.

Per ulteriori informazioni, consulta Monitoraggio dei parametri di S3 Storage Lens in CloudWatch.

Tutte le CloudWatch statistiche vengono conservate per un periodo di 15 mesi in modo da poter accedere alle informazioni storiche e avere una prospettiva migliore sulle prestazioni dell'applicazione o del servizio web. Per ulteriori informazioni su CloudWatch, consulta What is Amazon CloudWatch? nella Amazon CloudWatch User Guide. Potrebbero essere necessarie alcune configurazioni aggiuntive per i tuoi CloudWatch allarmi, a seconda dei casi d'uso. Ad esempio, puoi utilizzare un'espressione matematica metrica per creare un allarme. Per ulteriori informazioni, consulta Use CloudWatch metrics, Use metric Math, Using Amazon CloudWatch alarms e Create a CloudWatch alarm based on a metric math expression nella Amazon User Guide. CloudWatch

Distribuzione dei parametri Best-Effort CloudWatch

CloudWatch le metriche vengono fornite con la massima diligenza possibile. La maggior parte delle richieste per un oggetto Amazon S3 con parametri di richiesta comporta l'invio di un punto dati a. CloudWatch

La completezza e la tempestività dei parametri non è garantita. È possibile che il data point per una richiesta specifica venga restituito con un timestamp successivo a quello del momento effettivo di elaborazione della richiesta. Il data point potrebbe subire un ritardo di un minuto prima di essere disponibile oppure potrebbe non essere consegnato affatto. CloudWatch CloudWatch le metriche delle richieste ti danno un'idea della natura del traffico rispetto alla media in tempo quasi reale. ma non intendono essere un resoconto completo di tutte le richieste.

Il fatto che questa funzione si basi sul miglior tentativo fa sì che i report disponibili nel pannello di controllo Fatturazione e gestione costi potrebbero includere una o più richieste di accesso non visibili nei parametri del bucket.

Per ulteriori informazioni, consulta i seguenti argomenti.

Argomenti

- Parametri e dimensioni
- Accesso alle metriche CloudWatch
- CloudWatch configurazioni delle metriche

Parametri e dimensioni

Le metriche e le dimensioni di storage che Amazon S3 invia ad CloudWatch Amazon sono elencate nelle seguenti tabelle.

Erogazione delle metriche Best CloudWatch Effort

CloudWatch le metriche vengono fornite con la massima diligenza possibile. La maggior parte delle richieste per un oggetto Amazon S3 con parametri di richiesta comporta l'invio di un punto dati a. CloudWatch

La completezza e la tempestività dei parametri non è garantita. È possibile che il data point per una richiesta specifica venga restituito con un timestamp successivo a quello del momento effettivo di elaborazione della richiesta. Il data point potrebbe subire un ritardo di un minuto prima di essere disponibile oppure potrebbe non essere consegnato affatto. CloudWatch CloudWatch le metriche delle richieste ti danno un'idea della natura del traffico rispetto alla media in tempo quasi reale. ma non intendono essere un resoconto completo di tutte le richieste.

Il fatto che questa funzione si basi sul miglior tentativo fa sì che i report disponibili nel pannello di controllo Fatturazione e gestione costi potrebbero includere una o più richieste di accesso non visibili nei parametri del bucket.

Argomenti

Parametri di storage giornalieri di Amazon S3 per i bucket in CloudWatch

Il namespace `AWS/S3` include i seguenti parametri di storage giornalieri per i bucket.

Parametro	Descrizione
`BucketSizeBytes`	La quantità di dati in byte archiviati in un bucket nelle seguenti classi di archiviazione: • Amazon S3 Standard (`STANDARD`) • Piano intelligente Amazon S3 (`INTELLIGENT_TIERING`) • Accesso Infrequente Amazon S3 Standard (`STANDARD_IA`) • S3 One Zone-Infrequent Access () `ONEZONE_IA` • Reduced Redundancy Storage (RRS) (`REDUCED_REDUNDANCY`) • Recupero istantaneo Amazon S3 Glacier (`GLACIER_IR`) • Deep Archive Amazon S3 Glacier (`DEEP_ARCHIVE`) • Recupero flessibile Amazon S3 Glacier (`GLACIER`) • S3 Express One Zone (`EXPRESS_ONEZONE`) Questo valore viene calcolato sommando le dimensioni di tutti gli oggetti e i metadati (come i nomi dei bucket) nel bucket (oggetti correnti e non correnti), inclusa la dimensione di tutte le parti per tutti i caricamenti multiparte incompleti nel bucket. Filtri validi per il tipo di archiviazione: `StandardStorage`, `IntelligentTieringFAStorage`, `IntelligentTieringIAStorage`, `IntelligentTieringAAStorage`, `IntelligentTieringAIAStorage`, `IntelligentTieringDAAStorage`, `StandardIAStorage`, `StandardIASizeOverhead`, `StandardIAObjectOverhead`, `OneZoneIAStorage`, `OneZoneIASizeOverhead`, `ReducedRedundancyStorage`, `GlacierInstantRetrievalSizeOverhead` `GlacierInstantRetrievalStorage`, `GlacierStorage`, `GlacierStagingStorage`, `GlacierObjectOverhead`, `GlacierS3ObjectOverhead`, `DeepArchiveStorage`, `DeepArchiveObjectOverhead`, `DeepArchiveS3ObjectOverhead`, `DeepArchiveStagingStorage` e `ExpressOneZone` (vedi la dimensione `StorageType`) Unità: byte

Parametro	Descrizione
	Statistiche valide: media
`NumberOfObjects`	Il numero totale di oggetti archiviati in un bucket a uso generico per tutte le classi di archiviazione. Questo valore è calcolato contando tutti gli oggetti nel bucket (sia oggetti correnti che non correnti), elimina marcatori e il numero totale di parti dei caricamenti in più parti incomplet i nel bucket. Per i bucket di directory con oggetti nella classe di archiviaz ione S3 Express One Zone, questo valore viene calcolato contando tutti gli oggetti nel bucket, ma non include caricamenti multipli incompleti nel bucket. Filtri validi per il tipo di storage: `AllStorageTypes` (vedi la dimension e `StorageType`) Unità: numero Statistiche valide: media

Parametri delle richieste Amazon S3 in CloudWatch

Il namespace `AWS/S3` include i seguenti parametri di richiesta. Questi parametri includono richieste non fatturabili (nel caso di richieste provenienti da e Replication`GET`). `CopyObject`

> **Note**
>
> I parametri delle richieste di Amazon S3 CloudWatch non sono supportati per i bucket di directory.

Parametro	Descrizione
`AllRequests`	Numero totale di richieste HTTP effettuate a un bucket Amazon S3, indipendentemente dal tipo. Se utilizzi la configurazione di una metrica con un filtro, questa metrica restituisce solo le richieste HTTP che soddisfano i requisiti del filtro.

Parametro	Descrizione
	Unità: numero Statistiche valide: somma
`GetRequests`	Numero totale di richieste HTTP `GET` effettuate per gli oggetti in un bucket Amazon S3. Non sono incluse le operazioni LIST. Questa metrica viene incrementata per l'origine di ogni richiesta. `CopyObject` Unità: numero Statistiche valide: somma **Note** Le richieste impaginate orientate all'elenco, come,, e altre ListMultipartUploadsListPartsListObjectVersions, non sono incluse in questa metrica.
`PutRequests`	Numero totale di richieste HTTP `PUT` effettuate per gli oggetti in un bucket Amazon S3. Questa metrica viene incrementata per la destinazi one di ogni richiesta. `CopyObject` Unità: numero Statistiche valide: somma
`DeleteRequests`	Numero totale di richieste HTTP `DELETE` effettuate per gli oggetti in un bucket Amazon S3. Questa metrica include anche le richieste. DeleteObj ects Questa metrica mostra il numero di richieste effettuate e non il numero di oggetti eliminati. Unità: numero Statistiche valide: somma

Parametro	Descrizione
`HeadRequests`	Numero di richieste HTTP `HEAD` effettuate su un bucket Amazon S3. Unità: numero Statistiche valide: somma
`PostRequests`	Numero di richieste HTTP `POST` effettuate su un bucket Amazon S3. Unità: numero Statistiche valide: somma **Note** DeleteObjectse SelectObjectContentle richieste non sono incluse in questa metrica.
`SelectRequests`	Il numero di SelectObjectContentrichieste Amazon S3 effettuate per oggetti in un bucket Amazon S3. Unità: numero Statistiche valide: somma
`SelectBytesScanned`	Il numero di byte di dati scansionati con le richieste Amazon SelectObjectContentS3 in un bucket Amazon S3. Unità: byte Statistiche valide: media (byte per richiesta), somma (byte per periodo), numero di esempi, minimo, massimo (come p100), qualsiasi percentile tra p0,0 e p99,9

Parametro	Descrizione
`SelectBytesReturned`	Il numero di byte di dati restituiti con le richieste Amazon SelectObjectContentS3 in un bucket Amazon S3. Unità: byte Statistiche valide: media (byte per richiesta), somma (byte per periodo), numero di esempi, minimo, massimo (come p100), qualsiasi percentile tra p0,0 e p99,9
`ListRequests`	Il numero di richieste HTTP che visualizzano l'elenco del contenuto di un bucket. Unità: numero Statistiche valide: somma
`BytesDownloaded`	Il numero di byte scaricati per le richieste effettuate a un bucket Amazon S3, in cui la risposta include un corpo. Unità: byte Statistiche valide: media (byte per richiesta), somma (byte per periodo), numero di esempi, minimo, massimo (come p100), qualsiasi percentile tra p0,0 e p99,9
`BytesUploaded`	Numero di byte caricati per le richieste effettuate su un bucket Amazon S3, in cui la richiesta include un corpo. Unità: byte Statistiche valide: media (byte per richiesta), somma (byte per periodo), numero di esempi, minimo, massimo (come p100), qualsiasi percentile tra p0,0 e p99,9

Parametro	Descrizione
`4xxErrors`	Il numero di richieste di codice di stato di errore del client HTTP 4 xx effettuate a un bucket Amazon S3 con un valore pari a 0 o 1. La statistic a Average (Media) mostra la frequenza degli errori, mentre la statistica Sum (Somma) mostra il conteggio per un tipo specifico di errore, per ogni periodo. Unità: numero Statistiche valide: media (rapporti per richiesta), somma (rapporti per periodo), minimo, massimo, numero di esempi
`5xxErrors`	Il numero di richieste di codice di stato di errore del server HTTP 5 xx effettuate a un bucket Amazon S3 con un valore pari a 0 o 1. La statistic a Average (Media) mostra la frequenza degli errori, mentre la statistica Sum (Somma) mostra il conteggio per un tipo specifico di errore, per ogni periodo. Unità: numero Statistiche valide: media (rapporti per richiesta), somma (rapporti per periodo), minimo, massimo, numero di esempi
`FirstByte Latency`	Per ogni richiesta, il tempo dalla ricezione della richiesta completa da parte di un bucket Amazon S3 all'inizio della restituzione della risposta. Unità: millisecondi Statistiche valide: Average (Media), Sum (Somma), Min (Minimo), Max (Massimo) (come p100), Sample Count (Numero di esempi), qualsiasi percentile tra p0,0 e p100

Parametro	Descrizione
`TotalRequestLatency`	Per ogni richiesta, il tempo trascorso dalla ricezione del primo byte all'invio dell'ultimo byte a un bucket Amazon S3, Questa metrica include il tempo necessario per ricevere il corpo della richiesta e inviare il corpo della risposta, non incluso in `FirstByteLatency` . Unità: millisecondi Statistiche valide: Average (Media), Sum (Somma), Min (Minimo), Max (Massimo) (come p100), Sample Count (Numero di esempi), qualsiasi percentile tra p0,0 e p100

Metriche di replica S3 in CloudWatch

Puoi monitorare l'avanzamento della replica con le metriche di replica S3 tramite il tracciamento dei byte in sospeso, delle operazioni in sospeso e della latenza di replica. Per ulteriori informazioni, consulta Monitoraggio dell'avanzamento con i parametri di replica.

> **Note**
>
> Puoi abilitare gli allarmi per i tuoi parametri di replica in Amazon. CloudWatch Quando configuri gli allarmi per i parametri di replica, imposta il campo Missing data treatment (Trattamento dati persi) su Treat missing data as ignore (maintain the alarm state) (Ignora i dati mancanti (stesso stato allarme)).

Parametro	Descrizione
`ReplicationLatency`	Il numero massimo di secondi con cui la destinazione della replica si Regione AWS trova indietro rispetto all'origine Regione AWS per una determinata regola di replica. Unità: secondi Statistiche valide: Max

Parametro	Descrizione
`BytesPendingReplication`	Numero totale di byte di oggetti in attesa di replica per una determinata regola di replica. Unità: byte Statistiche valide: Max
`OperationsPendingReplication`	Numero di operazioni in attesa di replica per una determinata regola di replica. Unità: numero Statistiche valide: Max
`OperationsFailedReplication`	Numero di operazioni che non sono state replicate per una determinata regola di replica. Unità: numero Statistiche valide: Sum (numero totale di operazioni non riuscite), Average (percentuale di errori), Sample Count (numero totale di operazioni di replica)

Metriche di S3 Storage Lens in CloudWatch

Puoi pubblicare i parametri di utilizzo e attività di S3 Storage Lens su Amazon CloudWatch per creare una visione unificata dello stato di salute operativo nei dashboard. CloudWatch Le metriche di S3 Storage Lens vengono pubblicate nel namespace in. `AWS/S3/Storage-Lens` CloudWatch L'opzione di CloudWatch pubblicazione è disponibile per i dashboard di S3 Storage Lens che sono stati aggiornati a metriche e consigli avanzati.

Per un elenco delle metriche di S3 Storage Lens pubblicate su, consulta. CloudWatch Glossario dei parametri di Amazon S3 Storage Lens Per un elenco completo delle dimensioni, consulta Dimensioni.

Metriche della richiesta S3 Object Lambda in CloudWatch

S3 Object Lambda include le seguenti metriche di richiesta.

Parametro	Descrizione
`AllRequests`	Numero totale di richieste HTTP effettuate su un bucket Amazon S3 utilizzando un punto di accesso Lambda per oggetti. Unità: numero Statistiche valide: somma
`GetRequests`	Numero totale di richieste HTTP `GET` effettuate per gli oggetti utilizzando un punto di accesso Lambda per oggetti. Questa metrica non include le operazioni LIST. Unità: numero Statistiche valide: somma
`BytesUploaded`	Numero totale di byte caricati in un bucket Amazon S3 utilizzando un punto di accesso Lambda per oggetti, in cui la richiesta include un corpo. Unità: byte Statistiche valide: media (byte per richiesta), somma (byte per periodo), numero di esempi, minimo, massimo (come p100), qualsiasi percentile tra p0,0 e p99,9
`PostRequests`	Numero di richieste HTTP `POST` effettuate in un bucket Amazon S3 utilizzando un punto di accesso Lambda per oggetti. Unità: numero Statistiche valide: somma
`PutRequests`	Numero di richieste HTTP `PUT` di oggetti effettuate in un bucket Amazon S3 utilizzando un punto di accesso Lambda per oggetti. Unità: numero Statistiche valide: somma

Parametro	Descrizione
`DeleteRequests`	Numero di richieste HTTP `DELETE` di oggetti effettuate in un bucket Amazon S3 utilizzando un punto di accesso Lambda per oggetti. Questa metrica include le richieste. DeleteObjects Questa metrica mostra il numero di richieste effettuate e non il numero di oggetti eliminati. Unità: numero Statistiche valide: somma
`BytesDownloaded`	Numero di byte scaricati per le richieste effettuate su un bucket Amazon S3 utilizzando un punto di accesso Lambda per oggetti, in cui la risposta include un corpo. Unità: byte Statistiche valide: media (byte per richiesta), somma (byte per periodo), numero di esempi, minimo, massimo (come p100), qualsiasi percentile tra p0,0 e p99,9
`FirstByte Latency`	Per ogni richiesta, il tempo dalla ricezione della richiesta completa da parte di un bucket Amazon S3 mediante il punto di accesso Lambda per oggetti all'inizio della restituzione della risposta. Questa metrica dipende dal tempo di esecuzione della funzione AWS Lambda per trasforma re l'oggetto prima che la funzione restituisca i byte al punto di accesso Lambda per oggetti. Unità: millisecondi Statistiche valide: Average (Media), Sum (Somma), Min (Minimo), Max (Massimo) (come p100), Sample Count (Numero di esempi), qualsiasi percentile tra p0,0 e p100

Parametro	Descrizione
`TotalRequestLatency`	Per ogni richiesta, il tempo trascorso dalla ricezione del primo byte all'invio dell'ultimo byte a un punto di accesso Lambda per oggetti. Questa metrica include il tempo necessario per ricevere il corpo della richiesta e inviare il corpo della risposta, non incluso in `FirstByteLatency` . Unità: millisecondi Statistiche valide: Average (Media), Sum (Somma), Min (Minimo), Max (Massimo) (come p100), Sample Count (Numero di esempi), qualsiasi percentile tra p0,0 e p100
`HeadRequests`	Numero di richieste HTTP `HEAD` effettuate in un bucket Amazon S3 utilizzando un punto di accesso Lambda per oggetti. Unità: numero Statistiche valide: somma
`ListRequests`	Numero di richieste HTTP `GET` che visualizzano l'elenco del contenuto di un bucket Amazon S3. Questa metrica include i byte delle operazioni `ListObjects` e `ListObjectsV2` . Unità: numero Statistiche valide: somma
`4xxErrors`	Il numero di richieste di codice di stato di errore del client HTTP 4 xx effettuate a un bucket Amazon S3 utilizzando un punto di accesso Object Lambda con un valore pari a 0 o 1. La statistica Average (Media) mostra la frequenza degli errori, mentre la statistica Sum (Somma) mostra il conteggio per un tipo specifico di errore, per ogni periodo. Unità: numero Statistiche valide: media (rapporti per richiesta), somma (rapporti per periodo), minimo, massimo, numero di esempi

Parametro	Descrizione
`5xxErrors`	Il numero di richieste di codice di stato di errore del server HTTP 5 xx effettuate a un bucket Amazon S3 utilizzando un punto di accesso Object Lambda con un valore pari a 0 o 1. La statistica Average (Media) mostra la frequenza degli errori, mentre la statistica Sum (Somma) mostra il conteggio per un tipo specifico di errore, per ogni periodo. Unità: numero Statistiche valide: media (rapporti per richiesta), somma (rapporti per periodo), minimo, massimo, numero di esempi
`ProxiedRequests`	Numero di richieste HTTP in un punto di accesso Lambda per oggetti che restituiscono la risposta standard dell'API Amazon S3. Tali richieste non hanno una funzione Lambda configurata. Unità: numero Statistiche valide: somma
`InvokedLambda`	Numero di richieste HTTP a un oggetto S3 in cui è stata richiamata una funzione Lambda. Unità: numero Statistiche valide: somma
`LambdaResponseRequests`	Numero totale di richieste `WriteGetObjectResponse` effettuate dalla funzione Lambda. Questa metrica si applica solo alle richieste `GetObject` .
`LambdaResponse4xx`	Il numero di errori del client HTTP 4 xx che si verificano quando si chiama `WriteGetObjectResponse` da una funzione Lambda. Questo parametro fornisce le stesse informazioni di `4xxErrors` , ma solo per le chiamate `WriteGetObjectResponse` .

Parametro	Descrizione
LambdaResponse5xx	Il numero di errori del server HTTP 5 xx che si verificano quando si chiama WriteGetObjectResponse da una funzione Lambda. Questo parametro fornisce le stesse informazioni di 5xxErrors , ma solo per le chiamate WriteGetObjectResponse .

Parametri di Amazon S3 on Outposts in CloudWatch

Per un elenco delle metriche utilizzate per i CloudWatch bucket S3 on Outposts, consulta. CloudWatch metriche

Dimensioni di Amazon S3 in CloudWatch

Le dimensioni elencate di seguito vengono utilizzate per filtrare i parametri Amazon S3.

Dimensione	Descrizione
BucketName	Questa dimensione filtra i dati richiesti solo per il bucket identificato.
StorageType	Questa dimensione filtra i dati archiviati in un bucket in base ai seguenti tipi di storage: • StandardStorage : numero di byte utilizzati per gli oggetti nella classe di archiviazione STANDARD. • IntelligentTieringAAStorage : numero di byte utilizzati per gli oggetti nel livello Archive Access (Accesso archiviazione) della classe di archiviazione INTELLIGENT_TIERING . • IntelligentTieringAIAStorage : numero di byte utilizzati per gli oggetti nel livello Archive Instant Access (Accesso di archiviazione immediato) della classe di archiviazione INTELLIGENT_TIERING . • IntelligentTieringDAAStorage : numero di byte utilizzati per gli oggetti nel livello Deep Archive Access

Dimensione	Descrizione
	(Accesso di archiviazione profonda) della classe di archiviazione `INTELLIGENT_TIERING` . • `IntelligentTieringFAStorage` : numero di byte utilizzati per gli oggetti nel livello Frequent Access (Accesso frequente) della classe di archiviazione `INTELLIGENT_TIERING` . • `IntelligentTieringIAStorage` : numero di byte utilizzati per gli oggetti nel livello Infrequent Access (Accesso infrequente) della classe di archiviazione `INTELLIGENT_TIERING` . • `StandardIAStorage` — Il numero di byte utilizzati per gli oggetti nella classe di storage S3 Standard-Infrequent Access (). `STANDARD_IA` • `StandardIASizeOverhead` : numero di byte utilizzati per gli oggetti di dimensioni inferiori a 128 KB nella classe di archiviazione `STANDARD_IA` . • `IntAAObjectOverhead` : per ogni oggetto nella classe di archiviazione `INTELLIGENT_TIERING` nel livello Archive Access (Accesso archiviazione), S3 Glacier aggiunge 32 KB di spazio di archiviazione per l'indice e i metadati correlati. Questi dati aggiuntivi sono necessari per identificare e ripristinare l'oggetto desiderato. Questo spazio di archiviazione aggiuntivo vene addebitato secondo le tariffe di S3 Glacier Flexible Retrieval. • `IntAAS3ObjectOverhead` : per ogni oggetto nella classe di archiviazione `INTELLIGENT_TIERING` nel livello Archive Access (Accesso archiviazione), Amazon S3 utilizza 8 KB di spazio di archiviazione per il nome dell'oggetto e altri metadati. Questo spazio di archiviazione aggiuntivo viene addebitato secondo le tariffe di S3 Standard. • `IntDAAObjectOverhead` : per ogni oggetto nella classe di archiviazione `INTELLIGENT_TIERING` nel livello Deep Archive Access (Accesso archiviazione profonda), S3 Glacier

Dimensione	Descrizione
	aggiunge 32 KB di spazio di archiviazione per l'indice e i metadati correlati. Questi dati aggiuntivi sono necessari per identificare e ripristinare l'oggetto desiderato. Questo storage aggiuntivo viene addebitato secondo le tariffe di storage di S3 Glacier Deep Archive. • `IntDAAS3ObjectOverhead` : per ogni oggetto nella classe di archiviazione `INTELLIGENT_TIERING` nel livello Deep Archive Access (Accesso archiviazione profonda), Amazon S3 aggiunge 8 KB di spazio di archiviazione per l'indice e i metadati correlati. Questi dati aggiuntivi sono necessari per identificare e ripristinare l'oggetto desiderato. Questo spazio di archiviazione aggiuntivo viene addebitato secondo le tariffe di S3 Standard. • `OneZoneIAStorage` : numero di byte utilizzati per gli oggetti nella classe di archiviazione S3 One Zone-Infrequent Access (Accesso Infrequente Amazon S3 OneZone) (`ONEZONE_IA`). • `OneZoneIASizeOverhead` : numero di byte utilizzati per gli oggetti di dimensioni inferiori a 128 KB nella classe di archiviazione `ONEZONE_IA` . • `ReducedRedundancyStorage` : il numero di byte utilizzat i per gli oggetti nella classe Reduced Redundancy Storage (RRS). • `GlacierInstantRetrievalSizeOverhead` : il numero di byte utilizzati per gli oggetti di dimensione inferiore a 128 KB nella classe di archiviazione S3 Glacier Instant Retrieval (Recupero istantaneo Amazon S3 Glacier). • `GlacierInstantRetrievalStorage` : il numero di byte utilizzati per gli oggetti nella classe di archiviazione S3 Glacier Instant Retrieval. • `GlacierStorage` : il numero di byte utilizzati per gli oggetti nella classe di archiviazione S3 Glacier Flexible Retrieval. • `GlacierStagingStorage` : numero di byte utilizzati per le parti di oggetti in più parti prima che la richiesta `CompleteM`

Dimensione	Descrizione
	ultipartUpload venga completata per gli oggetti nella classe di archiviazione S3 Glacier Flexible Retrieval (Recupero flessibile Amazon S3 Glacier). • GlacierObjectOverhead : per ogni oggetto archiviato, S3 Glacier aggiunge 32 KB di spazio di archiviazione per l'indice e i metadati correlati. Questi dati aggiuntivi sono necessari per identificare e ripristinare l'oggetto desiderato. Questo spazio di archiviazione aggiuntivo vene addebitato secondo le tariffe di S3 Glacier Flexible Retrieval. • GlacierS3ObjectOverhead : per ogni oggetto archiviato in S3 Glacier Flexible Retrieval, Amazon S3 utilizza 8 KB di spazio di archiviazione per il nome dell'oggetto e gli altri metadati. Questo spazio di archiviazione aggiuntivo viene addebitato secondo le tariffe di S3 Standard. • DeepArchiveStorage : il numero di byte utilizzati per gli oggetti nella classe di archiviazione S3 Glacier Deep Archive. • DeepArchiveObjectOverhead : per ogni oggetto archiviato, S3 Glacier aggiunge 32 KB di spazio di archiviazione per l'indice e i metadati correlati. Questi dati aggiuntivi sono necessari per identificare e ripristinare l'oggetto desiderato. Questo storage aggiuntivo viene addebitato secondo le tariffe di S3 Glacier Deep Archive. • DeepArchiveS3ObjectOverhead : per ogni oggetto archiviato in S3 Glacier Deep Archive, Amazon S3 utilizza 8 KB di spazio di archiviazione per il nome dell'oggetto e gli altri metadati. Questo spazio di archiviazione aggiuntivo viene addebitato secondo le tariffe di S3 Standard. • DeepArchiveStagingStorage : numero di byte utilizzati per parti di oggetti in più parti prima che la richiesta CompleteMultipartUpload venga completata in oggetti nella classe di archiviazione S3 Glacier Deep Archive (Archiviazione profonda Amazon S3 Glacier).

Dimensione	Descrizione
	• `ExpressOneZone` : numero di byte utilizzati per gli oggetti nella classe di archiviazione S3 Express One Zone.
`FilterId`	Questa dimensione filtra le configurazioni delle metriche specificate per le metriche delle richieste in un bucket. Quando crei una configurazione delle metriche, specifichi un ID filtro (ad esempio, un prefisso, un tag o un punto di accesso). Per ulteriori informazioni, consulta la sezione Creazione di una configurazione dei parametri.

Dimensioni della replica S3 in CloudWatch

Le seguenti dimensioni vengono utilizzate per filtrare le metriche di replica S3.

Dimensione	Descrizione
`SourceBucket`	Il nome degli oggetti bucket da cui vengono replicati.
`DestinationBucket`	Il nome degli oggetti bucket in cui vengono replicati.
`RuleId`	Un identificatore univoco per la regola che ha attivato l'aggiornamento di questa metrica di replica.

Dimensioni di S3 Storage Lens in CloudWatch

Per un elenco delle dimensioni utilizzate per filtrare le metriche di S3 Storage Lens, consulta CloudWatch. Dimensioni

Dimensioni della richiesta S3 Object Lambda in CloudWatch

Le dimensioni riportate di seguito vengono utilizzate per filtrare i dati in un punto di accesso Lambda per oggetti.

Dimensione	Descrizione
`AccessPointName`	Nome del punto di accesso a cui vengono effettuate le richieste.

Dimensione	Descrizione
`DataSourceARN`	Origine da cui il punto di accesso Lambda per oggetti sta recuperando i dati. Se la richiesta richiama una funzione Lambda, si riferisce al Lambda Amazon Resource Name (ARN). Altrimenti si riferisce all'ARN del punto di accesso.

Accesso alle metriche CloudWatch

È possibile utilizzare le seguenti procedure per visualizzare le metriche di archiviazione per Amazon S3. Per includere i parametri di Amazon S3, è necessario impostare un timestamp di inizio e uno di fine. Per i parametri relativi a un periodo specifico di 24 ore, impostare il periodo di tempo su 86400 secondi, ovvero il numero di secondi in un giorno. Ricordare anche di impostare le dimensioni `BucketName` e `StorageType`.

Usando il AWS CLI

Ad esempio, se desideri utilizzare il per AWS CLI ottenere la media della dimensione di un bucket specifico in byte, puoi usare il seguente 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=DOC-EXAMPLE-BUCKET
 Name=StorageType,Value=StandardStorage --period 86400 --output json
```

Questo esempio produce il seguente output.

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-19T00:00:00Z",
            "Average": 1025328.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "BucketSizeBytes"
}
```

Utilizzo della console S3

Per visualizzare le metriche utilizzando la console Amazon CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione a sinistra scegli Metrics (Parametri).
3. Scegliere il namespace S3.
4. (Facoltativo) Per visualizzare un parametro, immettere il nome parametro nella casella di ricerca.
5. (Facoltativo) Per filtrare in base alla StorageTypedimensione, inserisci il nome della classe di archiviazione nella casella di ricerca.

Per visualizzare un elenco di metriche valide memorizzate per te Account AWS utilizzando il AWS CLI

- Al prompt dei comandi utilizza il comando seguente.

```
aws cloudwatch list-metrics --namespace "AWS/S3"
```

Per ulteriori informazioni sulle autorizzazioni necessarie per accedere alle CloudWatch dashboard, consulta le autorizzazioni della CloudWatch dashboard di Amazon nella Amazon CloudWatch User Guide.

CloudWatch configurazioni delle metriche

Con Amazon CloudWatch Request Metrics for Amazon S3, puoi ricevere parametri di CloudWatch 1 minuto, CloudWatch impostare allarmi e CloudWatch accedere a dashboard per near-real-time visualizzare le operazioni e le prestazioni del tuo storage Amazon S3. Per le applicazioni che dipendono dallo storage nel cloud, questi parametri consentono di identificare rapidamente i problemi operativi e intraprendere le azioni appropriate. Quando sono abilitati, questi parametri da 1 minuto sono disponibili a livello di bucket Amazon S3 per default.

Se desideri ottenere i parametri di CloudWatch richiesta per gli oggetti in un bucket, devi creare una configurazione dei parametri per il bucket. Per ulteriori informazioni, consulta Creazione di una configurazione CloudWatch delle metriche per tutti gli oggetti nel bucket.

Puoi anche definire un filtro per i parametri raccolti usando un prefisso condiviso, tag di oggetto o un punto di accesso. Questo metodo di definizione di un filtro consente di allineare i filtri dei parametri a

determinati flussi di lavoro, applicazioni business o organizzazioni interne. Per ulteriori informazioni, consulta Creazione di una configurazione dei parametri che filtra in base al prefisso, al tag oggetto o al punto di accesso. Per ulteriori informazioni sulle CloudWatch metriche disponibili e sulle differenze tra le metriche di archiviazione e di richiesta, consulta. Monitoraggio delle metriche con Amazon CloudWatch

Quando si usano le configurazioni dei parametri, tenere presente quanto indicato di seguito:

- È possibile definire un massimo di 1.000 configurazioni dei parametri per ciascun bucket.
- È possibile scegliere quali oggetti di un bucket includere nelle configurazioni dei parametri utilizzando i filtri. L'applicazione di un filtro utilizzando un prefisso condiviso, un tag di oggetto o un punto di accesso consente di allineare i filtri dei parametri a determinati flussi di lavoro, applicazioni business o organizzazioni interne. Per richiedere i parametri per l'intero bucket, creare una configurazione di parametro senza filtri.
- Le configurazioni dei parametri sono necessarie solo per abilitare i parametri di richiesta. I parametri di storage giornalieri a livello di bucket sono sempre attivi e disponibili senza costi aggiuntivi. Attualmente, non è possibile ottenere parametri di storage giornalieri per un sottoinsieme filtrato di oggetti.
- Ogni configurazione di parametro abilita l'intero insieme di parametri di richiesta disponibili. I parametri specifici delle operazioni (come ad esempio `PostRequests`) vengono indicati solo in presenza di richieste di quel tipo per il bucket o il filtro.
- I parametri della richiesta vengono indicati per le operazioni a livello di oggetto. Vengono indicati anche per le operazioni che elencano i contenuti del bucket, come GET Bucket (List Objects) (Elenca oggetti GET Bucket), GET Bucket Object Versions (Versioni dell'oggetto GET Bucket) e List Multipart Uploads (Elenca caricamenti in più parti), ma non vengono indicati per altre operazioni sui bucket.
- I parametri di richiesta supportano i filtri in base al prefisso, al tag oggetto o al punto di accesso, diversamente dai parametri di storage.

Fornitura delle metriche CloudWatch Best-Effort

CloudWatch le metriche vengono fornite con la massima diligenza possibile. La maggior parte delle richieste per un oggetto Amazon S3 con parametri di richiesta comporta l'invio di un punto dati a. CloudWatch

La completezza e la tempestività dei parametri non è garantita. È possibile che il data point per una richiesta specifica venga restituito con un timestamp successivo a quello del momento effettivo di

elaborazione della richiesta. Il data point potrebbe subire un ritardo di un minuto prima di essere disponibile oppure potrebbe non essere consegnato affatto. CloudWatch CloudWatch le metriche delle richieste ti danno un'idea della natura del traffico rispetto alla media in tempo quasi reale. ma non intendono essere un resoconto completo di tutte le richieste.

Il fatto che questa funzione si basi sul miglior tentativo fa sì che i report disponibili nel pannello di controllo Fatturazione e gestione costi potrebbero includere una o più richieste di accesso non visibili nei parametri del bucket.

Per ulteriori informazioni sull'utilizzo dei CloudWatch parametri in Amazon S3, consulta i seguenti argomenti.

Argomenti

- Creazione di una configurazione CloudWatch delle metriche per tutti gli oggetti nel bucket
- Creazione di una configurazione dei parametri che filtra in base al prefisso, al tag oggetto o al punto di accesso
- Eliminazione di un filtro dei parametri

Creazione di una configurazione CloudWatch delle metriche per tutti gli oggetti nel bucket

Quando configuri le metriche di richiesta, puoi creare una configurazione di CloudWatch metriche per tutti gli oggetti nel tuo bucket oppure puoi filtrare per prefisso, tag di oggetto o punto di accesso. Nelle procedure descritte in questo argomento viene illustrato come creare una configurazione per tutti gli oggetti nel bucket. Per creare una configurazione che filtri in base al tag oggetto, al prefisso o al punto di accesso, consulta Creazione di una configurazione dei parametri che filtra in base al prefisso, al tag oggetto o al punto di accesso.

Esistono tre tipi di CloudWatch parametri Amazon per Amazon S3: parametri di storage, parametri di richiesta e parametri di replica. I parametri di storage vengono indicati una volta al giorno e sono disponibili per tutti i clienti senza costi aggiuntivi. I parametri di richiesta sono disponibili a intervalli di 1 minuto dopo una determinata latenza per l'elaborazione. I parametri delle richieste vengono fatturati alla tariffa standard. CloudWatch È necessario acconsentire esplicitamente ai parametri di richiesta configurandoli nella console o utilizzando l'API Amazon S3. Le metriche di replica S3 forniscono metriche dettagliate per le regole nella configurazione di replica. Con le metriche di replica, è possibile monitorare l' minute-by-minute avanzamento tenendo traccia dei byte in sospeso, delle operazioni in sospeso, delle operazioni che non hanno avuto esito positivo e della latenza di replica.

Per ulteriori informazioni sui CloudWatch parametri per Amazon S3, consulta. Monitoraggio delle metriche con Amazon CloudWatch

Puoi aggiungere configurazioni dei parametri a un bucket utilizzando la console di Amazon S3, la AWS Command Line Interface (AWS CLI) o REST API di Amazon S3.

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket name (Nome bucket) selezionare il nome del bucket contenente gli oggetti di cui si desidera ottenere i parametri.
3. Seleziona la scheda Parametri.
4. In Bucket metrics (Parametri bucket) scegliere View additional charts (Visualizza grafici aggiuntivi).
5. Scegliere la scheda Request metrics (Parametri di richiesta).
6. Scegliere Create Filter (Crea filtro).
7. Nella casella Filter name (Nome filtro) immettere il nome del filtro.

   I nomi possono contenere solo lettere, numeri, punti, trattini e caratteri di sottolineatura. Si consiglia di utilizzare il nome `EntireBucket` per un filtro che si applica a tutti gli oggetti.
8. In Filter scope (Ambito del filtro) seleziona This filter applies to all objects in the bucket (Questo filtro si applica a tutti gli oggetti del bucket).

   È anche possibile definire un filtro in modo che i parametri vengano acquisiti e indicati solo per un sottoinsieme di oggetti del bucket. Per ulteriori informazioni, consulta Creazione di una configurazione dei parametri che filtra in base al prefisso, al tag oggetto o al punto di accesso.
9. Seleziona Save changes (Salva modifiche).
10. Nella scheda Request metrics (Parametri di richiesta), in Filters (Filtri), scegliere il filtro appena creato.

    Dopo circa 15 minuti, CloudWatch inizia a tracciare questi parametri di richiesta. Puoi visualizzarli nella scheda Request metrics (Parametri di richiesta) . Puoi visualizzare i grafici delle metriche su Amazon CloudWatch S3 o sulla console. I parametri delle richieste vengono fatturati alla tariffa standard. CloudWatch Per ulteriori informazioni, consulta i CloudWatch prezzi di Amazon.

Utilizzo di REST API

È anche possibile aggiungere configurazioni di parametri in modo programmatico con REST API di Amazon S3. Per ulteriori informazioni sull'aggiunta e sull'utilizzo delle configurazioni dei parametri, consulta i seguenti argomenti nella Documentazione di riferimento delle API di Amazon Simple Storage Service:

- Configurazione del parametro PUT Bucket
- Configurazione del parametro GET Bucket
- Configurazione del parametro LIST Bucket
- Configurazione del parametro DELETE Bucket

Utilizzando il AWS CLI

1. Installa e configura il AWS CLI. Per le istruzioni, consulta Installazione, aggiornamento e disinstallazione della AWS CLI nella Guida per l'utente di AWS Command Line Interface .
2. Aprire un terminale.
3. Eseguire il comando riportato di seguito per aggiungere una configurazione di parametro.

```
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"}'
```

Creazione di una configurazione dei parametri che filtra in base al prefisso, al tag oggetto o al punto di accesso

Esistono tre tipi di CloudWatch parametri Amazon per Amazon S3: parametri di storage, parametri di richiesta e parametri di replica. I parametri di storage vengono indicati una volta al giorno e sono disponibili per tutti i clienti senza costi aggiuntivi. I parametri di richiesta sono disponibili a intervalli di 1 minuto dopo una determinata latenza per l'elaborazione. I parametri delle richieste vengono fatturati alla tariffa standard. CloudWatch È necessario acconsentire esplicitamente ai parametri di richiesta configurandoli nella console o utilizzando l'API Amazon S3. Le metriche di replica S3 forniscono metriche dettagliate per le regole nella configurazione di replica. Con le metriche di replica, è possibile monitorare l' minute-by-minute avanzamento tenendo traccia dei byte in sospeso, delle operazioni in sospeso, delle operazioni che non hanno avuto esito positivo e della latenza di replica.

Per ulteriori informazioni sui CloudWatch parametri per Amazon S3, consulta. Monitoraggio delle metriche con Amazon CloudWatch

Quando configuri le CloudWatch metriche, puoi creare un filtro per tutti gli oggetti nel tuo bucket oppure puoi filtrare la configurazione in gruppi di oggetti correlati all'interno di un singolo bucket. Puoi filtrare gli oggetti in un bucket da includere in una configurazione di parametro in base a uno o più dei tipi di filtro elencati di seguito.

- Prefisso del nome della chiave dell'oggetto: sebbene il modello di dati Amazon S3 abbia una struttura orizzontale, è possibile applicare una gerarchia utilizzando un prefisso. La console di Amazon S3 supporta questi prefissi con il concetto di cartelle. Se si applica un filtro basato sul prefisso, gli oggetti con lo stesso prefisso verranno inclusi nella configurazione del parametro. Per ulteriori informazioni sui prefissi, consulta Organizzazione degli oggetti utilizzando i prefissi.
- Tag - È possibile aggiungere tag, che sono coppie di nomi chiave-valore, agli oggetti. I tag consentono di trovare e organizzare gli oggetti in modo semplice È possibile utilizzare i tag anche come filtri per le configurazioni dei parametri. Per ulteriori informazioni sui tag degli oggetti, consulta Suddivisione in categorie dello storage utilizzando i tag.
- Punto di accesso: i punti di accesso S3 sono endpoint di rete denominati e allegati a bucket che semplificano la gestione degli accessi ai dati su vasta scala per set di dati condivisi in S3. Quando crei un filtro in base al punto di accesso, Amazon S3 include le richieste al punto di accesso specificato nella configurazione dei parametri. Per ulteriori informazioni, consulta Monitoraggio e registrazione degli access point.

> **Note**
>
> Quando crei una configurazione di parametri che filtra in base al punto di accesso, devi utilizzare l'Amazon Resource Name (ARN) del punto di accesso e non l'alias del punto di accesso. Assicurati di utilizzare l'ARN del punto di accesso e non l'ARN di un oggetto specifico. Per ulteriori informazioni sugli ARN dei punti di accesso, consulta la sezione Utilizzo degli access point.

Se si specifica un filtro, solo le richieste che agiscono su oggetti singoli possono corrispondere al filtro ed essere incluse nei parametri dichiarati. Richieste simili DeleteObjectse `ListObjects` richieste non restituiscono alcuna metrica per le configurazioni con filtri.

Per richiedere un'applicazione più complessa di filtri, scegliere due o più elementi. Nella configurazione del parametro verranno inclusi solo gli oggetti con tutti questi elementi. Se non si impostano i filtri, tutti gli oggetti nel bucket vengono inclusi nella configurazione del parametro.

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegli il nome del bucket contenente gli oggetti di cui desideri ottenere i parametri.
3. Seleziona la scheda Parametri.
4. In Bucket metrics (Parametri bucket) scegliere View additional charts (Visualizza grafici aggiuntivi).
5. Scegliere la scheda Request metrics (Parametri di richiesta).
6. Scegliere Create Filter (Crea filtro).
7. Nella casella Filter name (Nome filtro) immettere il nome del filtro.

   I nomi possono contenere solo lettere, numeri, punti, trattini e caratteri di sottolineatura.
8. In Filter scope (Ambito del filtro), scegli Limit the scope of this filter using a prefix, object tags, and an S3 Access Point, or a combination of all three (Limita l'ambito di questo filtro utilizzando un prefisso, tag oggetto e un punto di accesso S3 o una combinazione dei tre).
9. In Filter type (Tipo di filtro), scegli almeno un tipo di filtro: Prefix (Prefisso),Object tags (Tag oggetto) oppure Access point (Punto di accesso).
10. Nella casella Prefix (Prefisso) inserisci un prefisso per definire un filtro in base al prefisso e limitare l'ambito del filtro a un singolo percorso.
11. Per definire un filtro in base ai tag oggetto, in Object tags (Tag oggetto), scegli Add tag (Aggiungi tag), quindi inserisci un tag Key (Chiave) e Value (Valore).
12. Per definire un filtro in base al punto di accesso, nel campo S3 Access point (Punto di accesso S3), inserisci l'ARN del punto di accesso o scegli Browse S3 (Sfoglia S3) per passare al punto di accesso.

> **Important**
>
> Non puoi inserire l'alias del punto di accesso. Devi inserire l'ARN del punto di accesso stesso e non l'ARN di un oggetto specifico.

13. Seleziona Salvataggio delle modifiche.

    Amazon S3 crea un filtro che utilizza il prefisso, i tag o il punto di accesso specificati.

14. Nella scheda Request metrics (Parametri di richiesta), in Filters (Filtri), scegliere il filtro appena creato.

    Hai creato un filtro che limita l'ambito dei parametri di richiesta in base al prefisso, ai tag oggetto o al punto di accesso. Circa 15 minuti dopo aver CloudWatch iniziato a tracciare questi parametri di richiesta, puoi visualizzare i grafici relativi ai parametri sia su Amazon CloudWatch S3 che sulle console. I parametri delle richieste vengono fatturati alla tariffa standard. CloudWatch Per ulteriori informazioni, consulta i CloudWatch prezzi di Amazon.

    È anche possibile richiedere i parametri a livello di bucket. Per informazioni, consulta Creazione di una configurazione CloudWatch delle metriche per tutti gli oggetti nel bucket.

Utilizzando il AWS CLI

1. Installa e configura il AWS CLI. Per le istruzioni, consulta Installazione, aggiornamento e disinstallazione della AWS CLI nella Guida per l'utente di AWS Command Line Interface .
2. Aprire un terminale.
3. Esegui uno dei comandi riportati di seguito per aggiungere una configurazione di parametri.

    Example : per filtrare in base al prefisso

```
aws s3api put-bucket-metrics-configuration --bucket DOC-EXAMPLE-BUCKET1 --
id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id", "Filter":
{"Prefix":"prefix1"}} '
```

    Example : per filtrare in base ai tag

```
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 : per filtrare in base al punto di accesso

```
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 : per filtrare in base al prefisso, ai tag e al punto di accesso

```
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"
        }
    }
}'
```

Utilizzo di REST API

È anche possibile aggiungere configurazioni di parametri in modo programmatico con REST API di Amazon S3. Per ulteriori informazioni sull'aggiunta e sull'utilizzo delle configurazioni dei parametri, consulta i seguenti argomenti nella Documentazione di riferimento delle API di Amazon Simple Storage Service:

- Configurazione del parametro PUT Bucket
- Configurazione del parametro GET Bucket
- Configurazione del parametro LIST Bucket
- Configurazione del parametro DELETE Bucket

Eliminazione di un filtro dei parametri

Puoi eliminare un filtro Amazon CloudWatch Request Metrics se non ti serve più. Quando elimini un filtro, non ti vengono più addebitati i parametri delle richieste che utilizzano quel filtro specifico. Tuttavia, le altre configurazioni di filtro esistenti continueranno a essere addebitate.

Quando si elimina un filtro, non è più possibile utilizzarlo per i parametri della richiesta. L'eliminazione di un filtro non può essere annullata.

Per informazioni sulla creazione di un filtro dei parametri delle richieste, consulta i seguenti argomenti:

- Creazione di una configurazione CloudWatch delle metriche per tutti gli oggetti nel bucket
- Creazione di una configurazione dei parametri che filtra in base al prefisso, al tag oggetto o al punto di accesso

Utilizzo della console S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket), scegliere il nome del bucket.
3. Seleziona la scheda Parametri.
4. In Bucket metrics (Parametri bucket) scegliere View additional charts (Visualizza grafici aggiuntivi).
5. Scegliere la scheda Request metrics (Parametri di richiesta).
6. Scegliere Manage filters (Gestisci filtri).
7. Scegliere il filtro.

> **Important**
>
> L'eliminazione di un filtro non può essere annullata.

8. Scegliere Delete (Elimina).

   Amazon S3 elimina il filtro.

Utilizzo di REST API

È anche possibile aggiungere configurazioni di parametri in modo programmatico con REST API di Amazon S3. Per ulteriori informazioni sull'aggiunta e sull'utilizzo delle configurazioni dei parametri, consulta i seguenti argomenti nella Documentazione di riferimento delle API di Amazon Simple Storage Service:

- Configurazione del parametro PUT Bucket
- Configurazione del parametro GET Bucket
- Configurazione del parametro LIST Bucket
- Configurazione del parametro DELETE Bucket

Notifiche di eventi Amazon S3

Puoi utilizzare la funzionalità Notifiche di eventi Amazon S3 per ricevere le notifiche relative a quando si verificano determinati eventi nel bucket S3. Per abilitare le notifiche, aggiungi una configurazione di notifica che identifichi gli eventi che Amazon S3 deve pubblicare. Assicurati inoltre che identifichi le destinazioni a cui Amazon S3 deve inviare le notifiche. Questa configurazione viene archiviata nella risorsa secondaria notifica associata a un bucket. Per ulteriori informazioni, consulta Opzioni di configurazione dei bucket. Amazon S3 fornisce un'API per la gestione di questa risorsa secondaria.

> **Important**
>
> Le notifiche degli eventi di Amazon S3 sono progettate per essere distribuite almeno una volta. Le notifiche di eventi in genere vengono distribuite in pochi secondi, ma a volte può essere necessario un minuto o più.

Panoramica delle notifiche eventi di Amazon S3.

Attualmente, Amazon S3 può pubblicare notifiche per i seguenti eventi:

- Eventi di creazione nuovo oggetto
- Eventi di rimozione di oggetti
- Eventi di ripristino di oggetti
- Un evento di perdita oggetto Reduced Redundancy Storage (RRS)

- Eventi di replica
- Eventi di scadenza del ciclo di vita S3
- Eventi di transizione del ciclo di vita S3
- Eventi di archiviazione automatica di S3 Intelligent-Tiering
- Eventi di assegnazione di tag agli oggetti
- Eventi di ACL PUT di oggetto

Per una descrizione completa dei tipi di evento, consulta Tipi di eventi supportati per SQS, SNS e Lambda.

Amazon S3 può inviare messaggi di notifica degli eventi alle seguenti destinazioni. Il valore dell'Amazon Resource Name (ARN) di queste destinazioni viene specificato nella configurazione della notifica.

- Argomenti su Amazon Simple Notification Service (Amazon SNS)
- Code di Amazon Simple Queue Service (Amazon SQS)
- AWS Lambda funzione
- Amazon EventBridge

Per ulteriori informazioni, consulta Destinazioni eventi supportate.

> **Note**
>
> Le code FIFO (First-In-First-Out) di Amazon Simple Queue Service non sono supportate come destinazione delle notifiche degli eventi di Amazon S3. Per inviare una notifica per un evento Amazon S3 a una coda FIFO di Amazon SQS, puoi utilizzare Amazon. EventBridge Per ulteriori informazioni, consulta Abilitare Amazon EventBridge.

> **Warning**
>
> Se la notifica scrive nello stesso bucket che attiva la notifica, potrebbe causare un loop di esecuzione. Ad esempio, se il bucket attiva una funzione Lambda ogni volta che un oggetto viene caricato e la funzione carica un oggetto nel bucket, allora la funzione indirettamente lo attiva. Per evitare questo, utilizzare due bucket, oppure configurare il trigger in modo che venga applicato solo a un prefisso utilizzato per gli oggetti in entrata.

Per ulteriori informazioni e un esempio di utilizzo delle notifiche di Amazon S3 con AWS Lambda, consulta AWS Lambda Using with Amazon S3 nella AWS Lambda Developer Guide.

Per ulteriori informazioni sul numero di configurazioni di notifica degli eventi che è possibile creare per bucket, consulta Quote di servizio Amazon S3 in Riferimenti generali di AWS .

Per ulteriori informazioni sulle notifiche degli eventi, consulta le sezioni seguenti.

Argomenti

Tipi di notifiche eventi e destinazioni

Amazon S3 supporta diversi tipi di notifiche di eventi e destinazioni in cui è possibile pubblicare le notifiche. Puoi specificare il tipo di evento e la destinazione durante la configurazione delle notifiche degli eventi. È possibile specificare una sola destinazione per ogni notifica di evento. Le notifiche degli eventi di Amazon S3 inviano una voce di evento per ogni messaggio di notifica.

Argomenti

Destinazioni eventi supportate

Amazon S3 può inviare messaggi di notifica degli eventi alle seguenti destinazioni.

- Argomenti su Amazon Simple Notification Service (Amazon SNS)
- Code di Amazon Simple Queue Service (Amazon SQS)
- AWS Lambda

- Amazon EventBridge

Tuttavia, è possibile specificare un solo tipo di destinazione per ogni notifica di evento.

> **Note**
>
> È necessario concedere ad Amazon S3; le autorizzazioni per pubblicare i messaggi in un argomento Amazon SNS o in una coda Amazon SQS. Devi inoltre concedere ad Amazon S3 l'autorizzazione a richiamare una AWS Lambda funzione per tuo conto. Per istruzioni su come concedere queste autorizzazioni, consulta Concessione di autorizzazioni per pubblicare messaggi di notifica eventi in una destinazione.

Argomento Amazon SNS

Amazon SNS è un servizio di messaggistica push completamente gestito e flessibile. Tramite questo servizio, è possibile inviare messaggi push a dispositivi mobili o servizi distribuiti. Con SNS puoi pubblicare un messaggio e inviarlo una o più volte. Al momento, l'SNS standard è consentito solo come destinazione di notifica di eventi S3, mentre l'SNS FIFO non è consentito.

Amazon SNS coordina e gestisce la consegna o l'invio di messaggi agli endpoint o ai client abbonati. È possibile utilizzare la console Amazon SNS per creare un argomento Amazon SNS a cui inviare le notifiche.

L'argomento deve essere nello Regione AWS stesso del bucket Amazon S3. Per informazioni sulla creazione di un argomento Amazon SNS, consulta Nozioni di base nella Guida per sviluppatori di Amazon Simple Notification Service e Domande frequenti su Amazon SNS.

Per poter utilizzare l'argomento Amazon SNS creato come destinazione della notifica di eventi, è necessario disporre di quanto segue:

- L'Amazon Resource Name (ARN) per l'argomento Amazon SNS
- Un'iscrizione valida a un argomento di Amazon SNS. Con esso, gli iscritti agli argomenti vengono informati quando un messaggio viene pubblicato sul tuo argomento Amazon SNS.

Coda Amazon SQS

Amazon SQS offre code ospitate affidabili e scalabili per lo storage dei messaggi mentre transitano tra computer. Tramite Amazon SQS è possibile trasmettere qualsiasi volume di dati senza richiedere

la disponibilità costante di altri servizi. È possibile utilizzare la console Amazon SQS per creare una coda Amazon SQS a cui inviare le notifiche.

La coda Amazon SQS deve trovarsi nella stessa Regione AWS posizione del bucket Amazon S3. Per informazioni sulla creazione di una coda di Amazon SQS, consulta Cos'è Amazon Simple Queue Service? e Nozioni di base su Amazon SQS nella Guida per sviluppatori di Amazon Simple Queue Service.

Per poter utilizzare la coda Amazon SQS come destinazione della notifica eventi, devi disporre di quanto segue:

- L'Amazon Resource Name (ARN) per la coda di Amazon SQS

> **Note**
>
> Le code FIFO (First-In-First-Out) di Amazon Simple Queue Service non sono supportate come destinazione delle notifiche degli eventi di Amazon S3. Per inviare una notifica per un evento Amazon S3 a una coda FIFO di Amazon SQS, puoi utilizzare Amazon. EventBridge Per ulteriori informazioni, consulta Abilitare Amazon EventBridge.

Funzione Lambda

Puoi utilizzarlo AWS Lambda per estendere altri AWS servizi con una logica personalizzata o creare un backend personalizzato che operi su AWS larga scala, con prestazioni e sicurezza. Con Lambda, è possibile creare applicazioni separate e basate su eventi che vengono eseguite solo quando necessario. È inoltre possibile utilizzarlo per dimensionare automaticamente queste applicazioni da poche richieste al giorno a migliaia al secondo.

Lambda esegue codice personalizzato in risposta agli eventi dei bucket Amazon S3. Il codice personalizzato viene caricato su Lambda e quindi viene creata quella che si chiama funzione Lambda. Quando Amazon S3 rileva un evento di un tipo specifico, può pubblicare l'evento AWS Lambda e richiamare la tua funzione in Lambda. In risposta, Lambda esegue la tua funzione. Un tipo di evento che potrebbe rilevare, ad esempio, è un evento di creazione oggetto.

Puoi usare la AWS Lambda console per creare una funzione Lambda che utilizza l' AWS infrastruttura per eseguire il codice per tuo conto. La funzione Lambda deve trovarsi nella stessa regione del bucket S3. Per impostare la funzione Lambda come destinazione di notifica eventi, è inoltre necessario disporre del nome o dell'ARN di una funzione Lambda.

> **Warning**
>
> Se la notifica scrive nello stesso bucket che attiva la notifica, potrebbe causare un loop di esecuzione. Ad esempio, se il bucket attiva una funzione Lambda ogni volta che un oggetto viene caricato e la funzione carica un oggetto nel bucket, allora la funzione indirettamente lo attiva. Per evitare questo, utilizzare due bucket, oppure configurare il trigger in modo che venga applicato solo a un prefisso utilizzato per gli oggetti in entrata.
> Per ulteriori informazioni e un esempio di utilizzo delle notifiche di Amazon S3 con AWS Lambda, consulta AWS Lambda Using with Amazon S3 nella AWS Lambda Developer Guide.

Amazon EventBridge

Amazon EventBridge è un bus di eventi senza server, che riceve eventi dai AWS servizi. Puoi impostare regole per abbinare gli eventi e distribuirli ai target, come ad esempio un servizio AWS o un endpoint HTTP. Per ulteriori informazioni, consulta Cosa c'è EventBridge nella Amazon EventBridge User Guide.

A differenza di altre destinazioni, puoi abilitare o disabilitare gli eventi a cui inviare eventi EventBridge per un bucket. Se abiliti la consegna, tutti gli eventi vengono inviati a EventBridge. Inoltre, puoi utilizzare EventBridge le regole per indirizzare gli eventi verso destinazioni aggiuntive.

Tipi di eventi supportati per SQS, SNS e Lambda

Amazon S3 pubblica i tipi di eventi riportati di seguito. Questi tipi di eventi devono essere specificati nella configurazione delle notifiche.

Tipi di eventi	Descrizione
s3: TestEvent	Quando una notifica è abilitata, Amazon S3 pubblica una notifica di prova. Questo serve a garantire che l'argomento esista e che il proprietario del bucket abbia l'autorizzazione a pubblicare l'argomento specificato. Se l'attivazione della notifica ha esito negativo, non verrà ricevuta alcuna notifica di prova.

Tipi di eventi	Descrizione
s3: * ObjectCreated s3:: Metti ObjectCreated s3:: Pubblica ObjectCreated s3:: Copia ObjectCreated s3:: ObjectCreated CompleteMultipartUpload	Le API di Amazon S3 come PUT, POST e COPY possono creare un oggetto. Con questi tipi di eventi, puoi abilitare le notifiche quando un oggetto viene creato tramite un'API specifica. In alternativa, puoi utilizzare il tipo di evento `s3:ObjectCreated:*` per richiedere la notifica indipendentemente dall'API utilizzata per creare un oggetto. `s3:ObjectCreated:CompleteMultipartUpload` include oggetti creati utilizzando UploadPartCopyper le operazioni di copia.
s3ObjectRemoved: * s3:: Elimina ObjectRemoved s3:: ObjectRemoved DeleteMarkerCreated	Utilizzando i tipi di ObjectRemovedeventi, è possibile abilitare la notifica quando un oggetto o un batch di oggetti viene rimosso da un bucket. È possibile richiedere una notifica quando viene eliminato un oggetto o quando viene eliminato in modo permanente e un oggetto con versione utilizzando il tipo di evento `s3:ObjectRemoved:Delete` . In alternativa, è possibile richiedere una notifica quando viene creato un contrassegno di eliminazione per un oggetto con versione utilizzando `s3:ObjectRemoved:DeleteMarkerCreated` . Per istruzioni su come eliminare gli oggetti con versione, consulta Eliminazione di versioni di oggetti da un bucket con funzione Controllo delle versioni abilitata . Inoltre, è possibile utilizzare un carattere jolly `s3:ObjectRemoved:*` per richiedere la notifica ogni volta che viene eliminato un oggetto. Queste notifiche di eventi non provvedono alcun avviso per le eliminazioni automatiche dalle configurazioni del ciclo di vita o dalle operazioni che hanno avuto esito negativo.

Tipi di eventi	Descrizione
s3: * ObjectRestore s3:: Pubblica ObjectRestore s3: :Completato ObjectRestore s3:: Elimina ObjectRestore	Utilizzando i tipi di ObjectRestoreeventi, è possibile ricevere notifiche per l'avvio e il completamento dell'evento durante il ripristino di oggetti dalla classe di storage S3 Glacier Flexible Retrieval, dalla classe di storage S3 Glacier Deep Archive, dal livello S3 Intelligent-Tiering Archive Access e dal livello S3 Intelligent-Tiering Deep Archive Access. È inoltre possibile ricevere notifiche per la scadenza della copia ripristinata di un oggetto. Il tipo di evento `s3:ObjectRestore:Post` notifica l'avvio del ripristino degli oggetti. Il tipo di evento `s3:ObjectRestore:Completed` notifica il completamento del ripristino. Il tipo di evento `s3:ObjectRestore:Delete` notifica la scadenza della copia temporanea di un oggetto ripristinato.
s3: ReducedRedundancyLostObject	Puoi ricevere questo evento di notifica quando Amazon S3 rileva che un oggetto della classe di archiviazione RRS è andato perso.

Tipi di eventi	Descrizione
s3:Replication:* s3: Replica: OperationFailedReplication S3: Replica: OperationMissedThreshold S3: Replica: OperationReplicatedAfterThreshold S3: Replica: OperationNotTracked	Con i tipi di evento Replication (Replica), è possibile ricevere notifiche di eventi per le configurazioni di replica con metriche di replica S3 o il controllo del tempo di replica di S3 (S3 RTC) abilitati. È possibile monitorare l' minute-by-minute avanzamento degli eventi di replica tenendo traccia dei byte in sospeso, delle operazioni in sospeso e della latenza di replica. Per informazioni sui parametri di replica, consulta Monitoraggio dell'avanzamento con le metriche di replica e le notifiche eventi di Amazon S3. Il tipo di evento `s3:Replication:OperationFailedReplication` notifica quando un oggetto idoneo per la replica non è stato replicato. Il tipo di evento `s3:Replication:OperationMissedThreshold` notifica quando un oggetto idoneo per la replica supera la soglia di 15 minuti per la replica. Il tipo di evento `s3:Replication:OperationReplicatedAfterThreshold` notifica quando un oggetto idoneo per la replica con S3 Replication Time Control viene replicato oltre il limite di 15 minuti. Il tipo di evento `s3:Replication:OperationNotTracked` notifica quando un oggetto idoneo per la replica con S3 Replication Time Control non è più monitorato dai parametri di replica.

Tipi di eventi	Descrizione
s3: * LifecycleExpiration s3:: Elimina LifecycleExpiration s3:: LifecycleExpiration DeleteMarkerCreated	Utilizzando i tipi di LifecycleExpirationeventi, puoi ricevere una notifica quando Amazon S3 elimina un oggetto in base alla tua configurazione del ciclo di vita S3. Il tipo di evento `s3:LifecycleExpiration:Delete` avvisa quando viene eliminato un oggetto in un bucket senza versione. Notifica inoltre quando una versione dell'oggetto viene eliminata definitivamente da una configurazione del ciclo di vita S3. Il tipo di evento `s3:LifecycleExpiration:DeleteMarkerCreated` notifica quando il ciclo di vita S3 crea un contrassegno di eliminazione quando viene eliminata una versione corrente di un oggetto in un bucket con versione.
s3: LifecycleTransition	Puoi ricevere questo evento di notifica quando un oggetto viene trasferito a un'altra classe di archiviazione Amazon S3 mediante una configurazione di S3 Lifecycle.
s3: IntelligentTiering	Puoi ricevere questo evento di notifica quando un oggetto all'interno della classe di archiviazione S3 Intelligent-Tiering viene spostato nel livello Archive Access o Deep Archive Access.
s3: * ObjectTagging s3:: Metti ObjectTagging s3:: Elimina ObjectTagging	Utilizzando i tipi di ObjectTaggingeventi, è possibile abilitare la notifica quando un tag di oggetto viene aggiunto o eliminato da un oggetto. Il tipo di evento `s3:ObjectTagging:Put` notifica quando un tag viene inserito su un oggetto tramite richiesta PUT o quando viene aggiornato un tag esistente. Il tipo di evento `s3:ObjectTagging:Delete` notifica quando un tag viene rimosso da un oggetto.

Tipi di eventi	Descrizione
s3:: Metti ObjectAcl	Puoi ricevere questo evento di notifica quando un'ACL viene inserita su un oggetto tramite richiesta PUT o quando viene modificata un'ACL esistente. Un evento non viene generato quando una richiesta non comporta alcuna modifica all'ACL di un oggetto.

Tipi di eventi supportati per Amazon EventBridge

Per un elenco dei tipi di eventi che Amazon S3 invierà ad Amazon EventBridge, consulta. Usando EventBridge

Ordinamento degli eventi ed eventi duplicati

Amazon S3 Event Notifications è progettato per inviare notifiche almeno una volta, ma non è garantito che arrivino nello stesso ordine in cui si sono verificati gli eventi. In rare occasioni, il meccanismo di riprova di Amazon S3 potrebbe causare notifiche di eventi S3 duplicate per lo stesso evento oggetto. Per ulteriori informazioni sulla gestione degli eventi duplicati o fuori servizio, consulta Manage event ordering and duplicate events with Amazon S3 Event Notifications sul blog di storage.AWS

Utilizzo di Amazon SQS, Amazon SNS e Lambda

L'abilitazione delle notifiche è un'operazione a livello di bucket. Le informazioni di configurazione delle notifiche vengono memorizzate nella risorsa secondaria di notifica associata a un bucket. Dopo avere creato o modificato la configurazione di notifica del bucket, in genere è necessario attendere 5 minuti affinché le modifiche abbiano effetto. Si verificherà un `s3:TestEvent` quando la notifica viene attivata per la prima volta. Per gestire la configurazione delle notifiche, è possibile utilizzare i metodi indicati di seguito:

- Utilizzo della console di Amazon S3 — E' possibile utilizzare l'interfaccia utente della console per impostare una configurazione di notifiche su un bucket senza dover scrivere alcun codice. Per ulteriori informazioni, consulta Attivazione e configurazione delle notifiche di eventi tramite la console di Amazon S3.
- Utilizzo programmatico degli AWS SDK: internamente, sia la console che gli SDK chiamano l'API REST di Amazon S3 per gestire le sottorisorse di notifica associate al bucket. Per esempi di

configurazione di notifiche che utilizzano gli SDK di AWS , consulta Spiegazione passo per passo: configurare un bucket per le notifiche (argomento SNS o coda SQS).

> **Note**
>
> Puoi effettuare le chiamate a REST API di Amazon S3 anche direttamente dal codice. Tuttavia, ciò può risultare scomodo in quanto richiede la scrittura di codice per autenticare le richieste.

Indipendentemente dal metodo utilizzato, Amazon S3 archivia la configurazione delle notifiche in formato XML nella risorsa secondaria notifica associata a un bucket. Per informazioni sulle risorse secondarie del bucket, consulta la sezione Opzioni di configurazione dei bucket.

Argomenti

- Concessione di autorizzazioni per pubblicare messaggi di notifica eventi in una destinazione
- Attivazione e configurazione delle notifiche di eventi tramite la console di Amazon S3
- Configurazione delle notifiche degli eventi a livello di programmazione
- Spiegazione passo per passo: configurare un bucket per le notifiche (argomento SNS o coda SQS)
- Configurazione delle notifiche di eventi mediante il filtro dei nomi delle chiavi oggetto
- Struttura del messaggio di evento

Concessione di autorizzazioni per pubblicare messaggi di notifica eventi in una destinazione

È necessario concedere al principale di Amazon S3 le autorizzazioni necessarie per richiamare l'API pertinente per pubblicare i messaggi in un argomento SNS, una coda SQS o una funzione Lambda. In questo modo Amazon S3 può pubblicare messaggi di notifica di eventi in una destinazione.

Per risolvere i problemi relativi alla pubblicazione dei messaggi di notifica di eventi su una destinazione, consulta Risoluzione dei problemi relativi alla pubblicazione delle notifiche di eventi di Amazon S3 in un argomento di Servizio di notifica semplice Amazon.

Argomenti

- Concessione delle autorizzazioni per richiamare una funzione AWS Lambda
- Concessione di autorizzazioni per pubblicare messaggi in un argomento SNS o una coda SQS

Concessione delle autorizzazioni per richiamare una funzione AWS Lambda

Amazon S3 pubblica messaggi di evento AWS Lambda invocando una funzione Lambda e fornendo il messaggio dell'evento come argomento.

Quando si utilizza la console di Amazon S3 per configurare le notifiche eventi in un bucket di Amazon S3 per una funzione Lambda, la console configura le autorizzazioni necessarie sulla funzione Lambda. In questo modo Amazon S3 dispone delle autorizzazioni per richiamare la funzione dal bucket. Per ulteriori informazioni, consulta Attivazione e configurazione delle notifiche di eventi tramite la console di Amazon S3.

Puoi anche concedere ad Amazon S3 le autorizzazioni AWS Lambda per richiamare la tua funzione Lambda. Per ulteriori informazioni, consulta Tutorial: Using AWS Lambda with Amazon S3 nella AWS Lambda Developer Guide.

Concessione di autorizzazioni per pubblicare messaggi in un argomento SNS o una coda SQS

Per concedere ad Amazon S3 le autorizzazioni per pubblicare messaggi sull'argomento SNS o sulla coda SQS, allega una policy AWS Identity and Access Management (IAM) all'argomento SNS o alla coda SQS di destinazione.

Per un esempio di come collegare una policy a un argomento SNS o a una coda SQS, consulta la sezione Spiegazione passo per passo: configurare un bucket per le notifiche (argomento SNS o coda SQS). Per ulteriori informazioni sulle autorizzazioni, consulta i seguenti argomenti:

- Casi di esempio per il controllo degli accessi ad Amazon SNS nella Guida per sviluppatori di Amazon Simple Notification Service
- Identity and Access Management in Amazon SQS nella Guida per gli sviluppatori di Amazon Simple Queue Service

Policy IAM per un argomento SNS di destinazione

Di seguito è riportato un esempio di policy AWS Identity and Access Management (IAM) da allegare all'argomento SNS di destinazione. Per maggiori informazioni su come utilizzare questa policy per configurare un argomento Amazon SNS di destinazione per le notifiche degli eventi, consulta Spiegazione passo per passo: configurare un bucket per le notifiche (argomento SNS o coda 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"
                }
            }
        }
    ]
}
```

Policy IAM per una coda SQS di destinazione

Di seguito è riportato un esempio di policy IAM collegata alla coda SQS di destinazione. Per maggiori informazioni su come utilizzare questa policy per impostare una coda Amazon SQS di destinazione per le notifiche degli eventi, consulta Spiegazione passo per passo: configurare un bucket per le notifiche (argomento SNS o coda SQS).

Per utilizzare questa politica, devi aggiornare l'ARN della coda Amazon SQS, il nome del bucket e l'ID del proprietario del bucket. Account AWS

```
{
    "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"
                }
            }
        }
    ]
}
```

Per entrambe le policy IAM di Amazon SNS e Amazon SQS, è possibile specificare la condizione `StringLike` nella policy anziché la condizione `ArnLike`.

Quando si utilizza `ArnLike`, le porzioni partizione, servizio, ID account, tipo di risorsa e ID risorsa parziale dell'ARN devono corrispondere esattamente all'ARN nel contesto della richiesta. La corrispondenza parziale è consentita solo per la regione e il percorso della risorsa.

Quando al posto di `StringLike` viene utilizzato `ArnLike`, la corrispondenza ignora la struttura dell'ARN e consente una corrispondenza parziale, indipendentemente dalla porzione utilizzata come carattere jolly. Per ulteriori informazioni, consulta Elementi delle policy JSON IAM nella Guida per l'utente di IAM.

```
"Condition": {
  "StringLike": { "aws:SourceArn": "arn:aws:s3:*:*:bucket-name" }
  }
```

AWS KMS politica chiave

Se la coda SQS o gli argomenti SNS sono crittografati con una chiave gestita dal cliente AWS Key Management Service (AWS KMS), devi concedere al servizio Amazon S3 l'autorizzazione principale per lavorare con gli argomenti o la coda crittografati. Per concedere al servizio Amazon S3 l'autorizzazione principale, aggiungi l'istruzione seguente alla policy delle chiavi per la chiave gestita dal 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": "*"
        }
    ]
}
```

Per ulteriori informazioni sulle politiche AWS KMS chiave, consulta Using key policy nella AWS KMS Developer Guide.AWS Key Management Service

Per ulteriori informazioni sull'utilizzo della crittografia lato server con AWS KMS Amazon SQS e Amazon SNS, consulta quanto segue:

- Gestione delle chiavi nella Guida per sviluppatori di Amazon Simple Notification Service.
- Gestione delle chiavi nella Guida per sviluppatori di Amazon Simple Queue Service.
- Crittografia dei messaggi pubblicati su Amazon SNS con AWS KMS nel Blog di calcolo di AWS .

Attivazione e configurazione delle notifiche di eventi tramite la console di Amazon S3

È possibile fare in modo che al verificarsi di determinati eventi di bucket Amazon S3 vengano inviati messaggi di notifica a una destinazione. In questa sezione viene descritto come utilizzare la console di Amazon S3 per abilitare le notifiche di evento. Per informazioni su come utilizzare le notifiche degli eventi con gli AWS SDK e le API REST di Amazon S3, consulta. Configurazione delle notifiche degli eventi a livello di programmazione

Prerequisiti: prima di abilitare le notifiche di eventi per il bucket, è necessario impostare uno dei tipi di destinazione e quindi configurare le autorizzazioni. Per ulteriori informazioni, consulta Destinazioni eventi supportate e Concessione di autorizzazioni per pubblicare messaggi di notifica eventi in una destinazione.

> **Note**
>
> Le code FIFO (First-In-First-Out) di Amazon Simple Queue Service non sono supportate come destinazione delle notifiche degli eventi di Amazon S3. Per inviare una notifica per un evento Amazon S3 a una coda FIFO di Amazon SQS, puoi utilizzare Amazon. EventBridge Per ulteriori informazioni, consulta Abilitare Amazon EventBridge.

Argomenti

Attivazione delle notifiche di Amazon SNS, Amazon SQS o Lambda tramite la console di Amazon S3

Per abilitare e configurare le notifiche di evento per un bucket S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket (Bucket) scegliere il nome del bucket per il quale abilitare gli eventi.
3. Scegliere Properties (Proprietà).
4. Passare alla sezione Event Notifications (Notifiche evento) e scegliere Create event notification (Crea notifica evento).
5. Nella sezione General configuration (Configurazione generale) specificare il nome dell'evento descrittivo per la notifica dell'evento. Facoltativamente, è anche possibile specificare un prefisso e un suffisso per limitare le notifiche agli oggetti con chiavi che terminano con i caratteri specificati.

   a. Immettere una descrizione per Event name (Nome evento).

      Se non specifichi un nome, viene generato e utilizzato un GUID (Globally Unique Identifier).

   b. (Facoltativo) Per filtrare le notifiche degli eventi in base al prefisso, immettere Prefix.

      Ad esempio, è possibile impostare un filtro prefisso in modo da ricevere notifiche solo quando i file vengono aggiunti a una cartella specifica co, `images/`).

   c. (Facoltativo) Per filtrare le notifiche degli eventi in base al suffisso, immettere Suffix.

Per ulteriori informazioni, consulta Configurazione delle notifiche di eventi mediante il filtro dei nomi delle chiavi oggetto.

6. Nella sezione Tipi di evento, selezionare uno o più tipi di evento per i quali si desidera ricevere notifiche.

   Per un elenco dei vari tipi di evento, consulta Tipi di eventi supportati per SQS, SNS e Lambda.

7. Nella sezione Destination (Destinazione), scegliere la destinazione della notifica dell'evento.

   > **Note**
   >
   > Prima di poter pubblicare le notifiche di eventi, è necessario concedere al principale di Amazon S3 le autorizzazioni necessarie per richiamare l'API pertinente. In questo modo è possibile pubblicare le notifiche su una funzione Lambda, un argomento SNS o una coda SQS.

   a. Selezionare il tipo di destinazione: Lambda Function (Funzione Lambda), SNS Topic (Argomento SNS) o SQS Queue (Coda SQS).
   b. Dopo aver scelto il tipo di destinazione, scegliere una funzione, un argomento o una coda dall'elenco.
   c. In alternativa, se preferisci specificare un Amazon Resource Name (ARN), seleziona Inserisci ARN e specifica l'ARN.

   Per ulteriori informazioni, consulta Destinazioni eventi supportate.

8. Seleziona Salva modifiche e Amazon S3 invia un messaggio di prova alla destinazione della notifica dell'evento.

Configurazione delle notifiche degli eventi a livello di programmazione

Per impostazione predefinita, le notifiche sono disattivate per tutti i tipi di evento. Pertanto, inizialmente la risorsa secondaria notifica archivia una configurazione vuota.

```
<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</NotificationConfiguration>
```

Per abilitare le notifiche per tipi di eventi specifici, sostituisci il file XML con la configurazione appropriata che identifica i tipi di evento che Amazon S3 deve pubblicare e la destinazione in cui gli eventi devono essere pubblicati. Per ciascuna destinazione, si aggiunge una configurazione XML corrispondente.

Per pubblicare messaggi di eventi in una coda SQS

Per impostare una coda SQS come destinazione di notifica per uno o più tipi di evento, aggiungi la `QueueConfiguration`.

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>optional-id-string</Id>
    <Queue>sqs-queue-arn</Queue>
    <Event>event-type</Event>
    <Event>event-type</Event>
     ...
  </QueueConfiguration>
   ...
</NotificationConfiguration>
```

Per pubblicare i messaggi di eventi in un argomento SNS

Per impostare un argomento SNS come destinazione di notifica per tipi di eventi specifici, aggiungi la `TopicConfiguration`.

```
<NotificationConfiguration>
  <TopicConfiguration>
     <Id>optional-id-string</Id>
     <Topic>sns-topic-arn</Topic>
     <Event>event-type</Event>
     <Event>event-type</Event>
      ...
  </TopicConfiguration>
   ...
</NotificationConfiguration>
```

Per richiamare la AWS Lambda funzione e fornire un messaggio di evento come argomento

Per impostare una funzione Lambda come destinazione di notifica per tipi di eventi specifici, aggiungi la `CloudFunctionConfiguration`.

```
<NotificationConfiguration>
  <CloudFunctionConfiguration>
     <Id>optional-id-string</Id>
     <CloudFunction>cloud-function-arn</CloudFunction>
     <Event>event-type</Event>
     <Event>event-type</Event>
      ...
  </CloudFunctionConfiguration>
   ...
</NotificationConfiguration>
```

Per rimuovere tutte le notifiche configurate su un bucket

Per rimuovere tutte le notifiche configurate in un bucket, salva un elemento `<NotificationConfiguration/>` vuoto nella risorsa secondaria di notifica.

Quando Amazon S3 rileva un evento di tipo specifico, pubblica un messaggio con le informazioni sull'evento. Per ulteriori informazioni, consulta Struttura del messaggio di evento.

Per ulteriori informazioni sulla configurazione delle notifiche di eventi, consulta i seguenti argomenti:

- Spiegazione passo per passo: configurare un bucket per le notifiche (argomento SNS o coda SQS).
- Configurazione delle notifiche di eventi mediante il filtro dei nomi delle chiavi oggetto

Spiegazione passo per passo: configurare un bucket per le notifiche (argomento SNS o coda SQS)

Puoi ricevere notifiche Amazon S3 utilizzando Amazon Simple Notification Service (Amazon SNS) o Amazon Simple Queue Service (Amazon SQS). Nella spiegazione passo per passo seguente viene aggiunta una configurazione di notifica al bucket utilizzando un argomento Amazon SNS e una coda Amazon SQS.

> **Note**
>
> Le code FIFO (First-In-First-Out) di Amazon Simple Queue Service non sono supportate come destinazione delle notifiche degli eventi di Amazon S3. Per inviare una notifica per un evento Amazon S3 a una coda FIFO di Amazon SQS, puoi utilizzare Amazon. EventBridge Per ulteriori informazioni, consulta Abilitare Amazon EventBridge.

Argomenti

Riepilogo della spiegazione passo per passo

Questa spiegazione passo per passo aiuta a completare le seguenti operazioni:

- Pubblicare eventi di tipo `s3:ObjectCreated:*` in una coda Amazon SQS.
- Pubblicare eventi di tipo `s3:ReducedRedundancyLostObject` in un argomento Amazon SNS.

Per informazioni sulla configurazione delle notifiche, consulta Utilizzo di Amazon SQS, Amazon SNS e Lambda.

È possibile eseguire tutte queste fasi utilizzando la console, senza scrivere alcun codice. Inoltre, vengono forniti anche esempi di codice che utilizzano AWS SDK per Java e.NET per aiutarti ad aggiungere configurazioni di notifica a livello di codice.

La procedura include le seguenti fasi:

1. Creare una coda Amazon SQS.

   Attraverso la console di Amazon SQS, crea una coda SQS. È possibile accedere a qualsiasi messaggio che Amazon S3 invia alla coda in modo programmatico. Tuttavia, per questa procedura guidata, i messaggi di notifica si verificano nella console.

   Collega una policy di accesso all'argomento per concedere ad Amazon S3 l'autorizzazione a pubblicare messaggi.
2. Creare un argomento Amazon SNS.

   Utilizzando la console di Amazon SNS, crea un argomento SNS e iscriviti all'argomento. In questo modo, riceverai tutti gli eventi pubblicati. Si specifica l'e-mail come protocollo di comunicazione. Dopo aver creato un argomento, Amazon SNS invia un'e-mail. È necessario utilizzare il collegamento nell'e-mail per confermare la sottoscrizione all'argomento.

Collega una policy di accesso all'argomento per concedere ad Amazon S3 l'autorizzazione a pubblicare messaggi.

3. Aggiungere una configurazione delle notifiche a un bucket.

Fase 1: creare una coda Amazon SQS

Segui le fasi per creare una coda Amazon Simple Queue Service (Amazon SQS) ed effettuarvi la sottoscrizione.

1. Utilizzando la console di Amazon SQS, creare una coda. Per istruzioni, consulta la sezione Nozioni di base su Amazon SQS nella Guida per gli sviluppatori di Amazon Simple Queue Service.
2. Sostituire la policy di accesso allegata alla coda con la policy riportata di seguito.

   a. Nella console di Amazon SQS, nell'elenco Code, seleziona il nome della coda.

   b. Nella scheda Policy di accesso, seleziona Modifica.

   c. Sostituire la policy di accesso allegata alla coda. Fornisci l'ARN di Amazon SQS, il nome del bucket di origine e l'ID dell'account del proprietario 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. Selezionare Salva.

3. (Facoltativo) Se la coda Amazon SQS o l'argomento Amazon SNS è la crittografia lato server abilitata con AWS Key Management Service (AWS KMS), aggiungi la seguente policy alla chiave di crittografia simmetrica associata gestita dal cliente.

   Devi aggiungere la policy a una chiave gestita dal cliente perché non è possibile modificare la chiave gestita da AWS per 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": "*"
        }
    ]
}
```

   Per ulteriori informazioni sull'utilizzo di SSE per Amazon SQS e Amazon SNS AWS KMS con, consulta quanto segue:

   - Gestione delle chiavi nella Guida per sviluppatori di Amazon Simple Notification Service.
   - Gestione delle chiavi nella Guida per sviluppatori di Amazon Simple Queue Service.

4. Prendere nota dell'ARN della coda.

La coda SQS creata è un'altra risorsa nel tuo Account AWS. Dispone di un Amazon Resource Name (ARN) univoco. Il presente ARN è necessario nella fase successiva. Il nome ARN presenta il formato seguente:

```
arn:aws:sqs:aws-region:account-id:queue-name
```

Fase 2: creare un argomento Amazon SNS

Completa la procedura per creare e sottoscrivere un argomento Amazon SNS.

1. Utilizzando la console di Amazon SNS, crea un argomento. Per le istruzioni, consulta la sezione Creazione di un argomento Amazon SNS nella Guida per gli sviluppatori di Amazon Simple Notification Service.
2. Effettuare la sottoscrizione all'argomento. Per questo esercizio, utilizzare l'e-mail come protocollo di comunicazione. Per le istruzioni, consulta la sezione Sottoscrizione a un argomento di Amazon SNS nella Guida per gli sviluppatori di Amazon Simple Notification Service.

   Si riceverà un'e-mail in cui è richiesto di confermare la sottoscrizione all'argomento. Confermare la sottoscrizione.
3. Sostituire la policy di accesso collegata all'argomento con la policy riportata di seguito. Fornisci l'ARN dell'argomento SNS, il nome del bucket e l'ID dell'account del proprietario 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. Prendere nota dell'ARN dell'argomento.

   L'argomento SNS che hai creato è un'altra tua Account AWS risorsa e ha un ARN unico. L'ARN è necessario nella fase successiva. L'ARN ha il formato seguente:

   ```
   arn:aws:sns:aws-region:account-id:topic-name
   ```

Fase 3: aggiungere una configurazione delle notifiche al bucket

Puoi abilitare le notifiche bucket utilizzando la console Amazon S3 o a livello di codice utilizzando gli SDK. AWS Scegliere una delle opzioni per configurare le notifiche nel bucket. Questa sezione fornisce esempi di codice che utilizzano gli SDK AWS per Java e .NET.

Opzione A: abilitare le notifiche in un bucket utilizzando la console

Utilizzando la console di Amazon S3, aggiungi una configurazione di notifica che richiede ad Amazon S3 di:

- Pubblicare gli eventi di tipo Tutti gli eventi di creazione dell'oggetto nella coda Amazon SQS.
- Pubblica gli eventi di tipo Oggetto perso in RRS sul tuo argomento Amazon SNS.

Una volta salvata la configurazione delle notifiche, Amazon S3 pubblica un messaggio di testo che viene inviato tramite e-mail.

Per istruzioni, consulta Attivazione e configurazione delle notifiche di eventi tramite la console di Amazon S3.

Opzione B: abilita le notifiche su un bucket utilizzando gli SDK AWS

.NET

L'esempio di codice C# riportato di seguito include il codice completo che aggiunge una configurazione delle notifiche in un bucket. Occorre aggiornare il codice e fornire il nome del bucket e l'ARN dell'argomento SNS. Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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

Nell'esempio seguente viene mostrato come aggiungere una configurazione delle notifiche a un bucket. Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
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();
        }
    }
}
```

Fase 4: eseguire il test della configurazione

Ora è possibile testare la configurazione caricando un oggetto nel bucket e verificando la notifica di eventi nella console di Amazon SQS. Per istruzioni, consulta la sezione Ricezione di un messaggio nella sezione "Nozioni di base" della Guida per gli sviluppatori di Amazon Simple Queue Service.

Configurazione delle notifiche di eventi mediante il filtro dei nomi delle chiavi oggetto

Quando configuri una notifica di eventi di Amazon S3, devi specificare quali tipi di eventi di Amazon S3 supportati fanno sì che Amazon S3 invii la notifica. Se nel bucket S3 si verifica un tipo di evento che non hai specificato, Amazon S3 non invia la notifica.

È possibile configurare le notifiche in modo da essere filtrate in base al prefisso e al suffisso del nome della chiave degli oggetti, Ad esempio, è possibile impostare una configurazione per ricevere una notifica solo quando i file di immagini con l'estensione del nome file "`.jpg`" vengono aggiunti a un bucket. In alternativa, puoi avere una configurazione che invia una notifica a un argomento di Amazon SNS quando un oggetto con il prefisso "`images/`" viene aggiunto al bucket, mentre le notifiche per gli oggetti con un prefisso "`logs/`" nello stesso bucket vengono inviate a una funzione. AWS Lambda

> **Note**
>
> Un carattere jolly ("*") non può essere utilizzato nei filtri come prefisso o suffisso. Se il prefisso o il suffisso contiene uno spazio, è necessario sostituirlo con il carattere "+". Se utilizzi altri caratteri speciali nel valore del prefisso o del suffisso, dovrai immetterli nel formato con codifica URL (codificato in percentuale). Per un elenco completo dei caratteri speciali che devono essere convertiti in formato con codifica URL se utilizzati in un prefisso o in un suffisso per le notifiche di eventi, consulta Caratteri sicuri.

È possibile impostare configurazioni delle notifiche che utilizzano il filtraggio del nome della chiave oggetto nella console di Amazon S3. Puoi farlo utilizzando le API di Amazon S3 tramite gli AWS SDK o le API REST direttamente. Per informazioni sull'utilizzo dell'interfaccia utente della console per impostare una configurazione di notifica in un bucket, consulta la sezione Attivazione e configurazione delle notifiche di eventi tramite la console di Amazon S3.

Amazon S3 archivia la configurazione delle notifiche in formato XML nella risorsa secondaria notification associata a un bucket, come descritto in Utilizzo di Amazon SQS, Amazon SNS e Lambda. Si utilizza la struttura XML `Filter` per definire le regole per filtrare le notifiche in base al prefisso o al suffisso del nome della chiave oggetto. Per informazioni sulla struttura XML `Filter`, consulta notifica PUT Bucket nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Le configurazioni delle notifiche che utilizzano `Filter` non possono definire regole di filtri con prefissi che si sovrappongono, suffissi che si sovrappongono o prefisso e suffisso che si sovrappongono. Nelle sezioni seguenti sono riportati esempi di configurazioni delle notifiche valide con il filtro del nome della chiave dell'oggetto. Contengono anche esempi di configurazioni delle notifiche non valide a causa della sovrapposizione di prefisso/suffisso.

Argomenti

- Esempi di configurazioni di notifiche valide con il filtro del nome della chiave dell'oggetto
- Esempi di configurazioni di notifiche con sovrapposizione di prefisso/suffisso non valida

Esempi di configurazioni di notifiche valide con il filtro del nome della chiave dell'oggetto

La configurazione delle notifiche seguente contiene una configurazione della coda che identifica una coda Amazon SQS dove Amazon S3 deve pubblicare gli eventi di tipo `s3:ObjectCreated:Put`. Gli eventi vengono pubblicati ogni volta che un oggetto con prefisso `images/` e suffisso `jpg` viene aggiunto a un bucket tramite richiesta PUT.

```
<NotificationConfiguration>
  <QueueConfiguration>
      <Id>1</Id>
      <Filter>
          <S3Key>
              <FilterRule>
                  <Name>prefix</Name>
                  <Value>images/</Value>
              </FilterRule>
```

```
            <FilterRule>
                <Name>suffix</Name>
                <Value>jpg</Value>
            </FilterRule>
        </S3Key>
    </Filter>
    <Queue>arn:aws:sqs:us-west-2:444455556666:s3notificationqueue</Queue>
    <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
</NotificationConfiguration>
```

La configurazione delle notifiche riportata di seguito include diversi prefissi che non si sovrappongono. La configurazione stabilisce che le notifiche delle richieste PUT nella cartella `images/` vanno nella coda A (queue-A) mentre le notifiche delle richieste PUT nella cartella `logs/` vanno nella coda B (queue-B).

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>1</Id>
    <Filter>
           <S3Key>
               <FilterRule>
                   <Name>prefix</Name>
                   <Value>images/</Value>
               </FilterRule>
           </S3Key>
    </Filter>
    <Queue>arn:aws:sqs:us-west-2:444455556666:sqs-queue-A</Queue>
    <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
  <QueueConfiguration>
    <Id>2</Id>
    <Filter>
           <S3Key>
               <FilterRule>
                   <Name>prefix</Name>
                   <Value>logs/</Value>
               </FilterRule>
           </S3Key>
    </Filter>
    <Queue>arn:aws:sqs:us-west-2:444455556666:sqs-queue-B</Queue>
    <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
```

```
</NotificationConfiguration>
```

La configurazione delle notifiche riportata di seguito include diversi suffissi che non si sovrappongono. La configurazione stabilisce che tutte le nuove immagini `.jpg` aggiunte al bucket verranno elaborate dalla funzione Lambda cloud-function-A, mentre tutte le nuove immagini `.png` verranno elaborate dalla funzione cloud-function-B. I suffissi `.png` e `.jpg` non si sovrappongono anche se hanno la stessa lettera finale. I due suffissi si considerano sovrapposti se una determinata stringa può terminare con entrambi. Una stringa non può terminare sia con `.png` che con `.jpg`, pertanto i suffissi nella configurazione di esempio non si sovrappongono.

```
<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>
```

Le configurazioni delle notifiche che utilizzano `Filter` non possono definire regole per filtrare i prefissi che si sovrappongono nello stesso tipo di evento. Possono farlo solo se i prefissi sovrapposti sono utilizzati con suffissi che non si sovrappongono. L'esempio di configurazione seguente mostra in che modo gli oggetti creati con un prefisso comune ma con suffissi che non si sovrappongono possono essere distribuiti in diverse destinazioni.

```
<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>
```

Esempi di configurazioni di notifiche con sovrapposizione di prefisso/suffisso non valida

La maggior parte delle configurazioni delle notifiche che utilizzano `Filter` non possono definire regole per filtrare i prefissi che si sovrappongono, i suffissi che si sovrappongono o le combinazioni di prefissi e suffissi che si sovrappongono per gli stessi tipi di eventi. È possibile avere prefissi che si sovrappongono a condizione che non si sovrappongano i suffissi. Per vedere un esempio, consulta Configurazione delle notifiche di eventi mediante il filtro dei nomi delle chiavi oggetto.

È possibile utilizzare filtri di nomi delle chiavi degli oggetti che si sovrappongono con diversi tipi di eventi. Ad esempio, si potrebbe creare una configurazione delle notifiche che utilizza il prefisso `image/` per il tipo di evento `ObjectCreated:Put` e il prefisso `image/` per il tipo di evento `ObjectRemoved:*`.

Se cerchi di salvare una configurazione delle notifiche con filtri di nomi non validi che si sovrappongono per gli stessi tipi di eventi, durante l'utilizzo dell'API o della console di Amazon S3, si verifica un errore. Questa sezione mostra esempi di configurazioni delle notifiche non valide a causa della sovrapposizione dei filtri dei nomi.

Si presuppone che qualsiasi regola di configurazione delle notifiche esistente abbia prefisso e suffisso predefiniti che corrispondano rispettivamente a qualsiasi altro prefisso e suffisso. La configurazione delle notifiche riportata di seguito non è valida poiché include prefissi che si sovrappongono. In particolare, il prefisso root si sovrappone con qualsiasi altro prefisso. Lo stesso vale nel caso in cui in questo esempio si utilizzi un suffisso anziché il prefisso. Il suffisso root si sovrappone a qualsiasi altro suffisso.

```
<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 configurazione delle notifiche riportata di seguito non è valida in quanto include suffissi che si sovrappongono. I due suffissi si considerano sovrapposti se una determinata stringa può terminare con entrambi. Una stringa può terminare con `jpg` e `pg`. Quindi, i suffissi si sovrappongono. Lo stesso vale per i prefissi. Due prefissi sono considerati sovrapposti se una determinata stringa può iniziare con entrambi i prefissi.

```
 <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 configurazione delle notifiche riportata di seguito non è valida poiché include prefissi e suffissi che si sovrappongono.

```
<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>
```

Struttura del messaggio di evento

Il messaggio di notifica inviato da Amazon S3 per pubblicare un evento è in formato JSON.

Per una panoramica generale e istruzioni sulla configurazione delle notifiche degli eventi, consulta Notifiche di eventi Amazon S3.

Questo esempio mostra la versione 2.2della struttura JSON di notifica degli eventi. Amazon S3 utilizza le versioni 2.1, 2.2 e 2.3 di questa struttura di eventi. Amazon S3 utilizza la versione 2.2 per le notifiche di eventi di replica tra Regioni. Utilizza la versione 2.3 per S3 Lifecycle, S3 Intelligent-Tiering, ACL di oggetti, assegnazione di tag di oggetti e ripristino oggetti per gli eventi di eliminazione. Queste versioni contengono informazioni aggiuntive specifiche per queste operazioni. Le versioni 2.2

e 2.3 sono altrimenti compatibili con la versione 2.1 che Amazon S3 utilizza attualmente per altri tipi di notifiche di eventi.

```
{
   "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"
            }
         }
      }
   ]
}
```

Notare quanto segue sulla struttura dei messaggi di evento:

- Il valore della chiave `eventVersion` contiene una versione maggiore e minore nel formato `<major>.<minor>`.

  La versione principale viene incrementata se Amazon S3 apporta una modifica alla struttura dell'evento che non è compatibile con le versioni precedenti. Questo include la rimozione di un campo JSON che è già presente o la modifica del modo in cui i contenuti di un campo vengono rappresentati (ad esempio, un formato di data).

  La versione secondaria viene incrementata se Amazon S3 aggiunge nuovi campi alla struttura dell'evento. Questo può succedere se vengono fornite nuove informazioni per alcuni o tutti gli eventi esistenti. Questo può succedere anche se vengono fornite nuove informazioni solo sui tipi di eventi appena introdotti. Le applicazioni devono ignorare i nuovi campi per rimanere compatibili con le nuove versioni minori della struttura dell'evento.

  Se vengono introdotti nuovi tipi di eventi ma la struttura dell'evento rimane invariata, la versione dell'evento non cambia.

  Per fare in modo che le applicazioni analizzino correttamente la struttura dell'evento, è consigliabile eseguire un confronto "uguale a" sul numero della versione maggiore. Per assicurarti che i campi previsti dall'applicazione siano presenti, ti consigliamo anche di fare un confronto greater-than-or-equal -to sulla versione secondaria.
- `eventName` fa riferimento all'elenco dei tipi di notifiche di eventi ma non contiene il prefisso `s3:`.
- Il valore `responseElements` chiave è utile se si desidera tracciare una richiesta dando seguito a AWS Support. Sia `x-amz-request-id` sia `x-amz-id-2` aiutano Amazon S3 a tenere traccia di una singola richiesta. Questi valori corrispondono a quelli che Amazon S3 restituisce nella risposta alla richiesta che avvia gli eventi. In questo modo, possono essere utilizzati per mettere in corrispondenza l'evento alla richiesta.

- La chiave `s3` fornisce informazioni sul bucket e sugli oggetti coinvolti nell'evento. Il valore del nome della chiave dell'oggetto ha la codifica URL. Ad esempio, "red flower.jpg" diventa "red +flower.jpg" (Amazon S3 restituisce "`application/x-www-form-urlencoded`" come tipo di contenuto nella risposta).
- La chiave `sequencer` fornisce un modo di stabilire la sequenza degli eventi. Non è garantito che le notifiche di eventi arrivino nello stesso ordine in cui avvengono gli eventi. Tuttavia, notifiche da eventi che creano oggetti (PUT) ed eliminano oggetti contengono un `sequencer`. Può essere utilizzato per determinare l'ordine degli eventi per una determinata chiave oggetto.

  Se si confrontano le stringhe `sequencer` da due notifiche eventi nella stessa chiave dell'oggetto, la notifica evento con il valore esadecimale `sequencer` più elevato è l'evento che è avvenuto per ultimo. Se si utilizzano notifiche di eventi per mantenere un database o un indice separato degli oggetti Amazon S3, è consigliabile confrontare e archiviare i valori `sequencer` man mano che la notifica di ciascun evento viene elaborata.

  Tieni presente quanto segue:

  - `sequencer` non può essere utilizzato per determinare l'ordine degli eventi su diverse chiavi dell'oggetto.
  - I sequencer possono essere di diversa lunghezza. Pertanto, per confrontare questi valori, occorre innanzitutto riempire il valore più corto con degli zeri, quindi eseguire un confronto lessicografico.
- La chiave `glacierEventData` è visibile solo per gli eventi `s3:ObjectRestore:Completed`.
- La chiave `restoreEventData` contiene attributi correlati alla richiesta di ripristino.
- La chiave `replicationEventData` è visibile solo per gli eventi di replica.
- La chiave `intelligentTieringEventData` è visibile solo per gli eventi S3 Intelligent-Tiering.
- La chiave `lifecycleEventData` è visibile solo per gli eventi di transizione del ciclo di vita S3.

Messaggi di esempio

Di seguito sono riportati alcuni esempi di messaggi di notifica degli eventi Amazon S3.

Messaggio di prova Amazon S3

Quando configuri una notifica di eventi in un bucket, Amazon S3 invia il messaggio di prova riportato di seguito.

```
{
```

```
   "Service":"Amazon S3",
   "Event":"s3:TestEvent",
   "Time":"2014-10-13T15:57:02.089Z",
   "Bucket":"bucketname",
   "RequestId":"5582815E1AEA5ADF",
   "HostId":"8cLeGAmw098X5cv4Zkwcmo8vvZa3eH3eKxsPzbB9wrR+YstdA6Knx4Ip8EXAMPLE"
}
```

Messaggio di esempio quando un oggetto viene creato utilizzando una richiesta PUT

Il seguente messaggio è un esempio di un messaggio inviato da Amazon S3 per pubblicare un evento `s3:ObjectCreated:Put`.

```
{
   "Records":[
      {
         "eventVersion":"2.1",
         "eventSource":"aws:s3",
         "awsRegion":"us-west-2",
         "eventTime":"1970-01-01T00:00:00.000Z",
         "eventName":"ObjectCreated:Put",
         "userIdentity":{
            "principalId":"AIDAJDPLRKLG7UEXAMPLE"
         },
         "requestParameters":{
            "sourceIPAddress":"127.0.0.1"
         },
         "responseElements":{
            "x-amz-request-id":"C3D13FE58DE4C810",
            "x-amz-id-2":"FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/
JRWeUWerMUE5JgHvANOjpD"
         },
         "s3":{
            "s3SchemaVersion":"1.0",
            "configurationId":"testConfigRule",
            "bucket":{
               "name":"mybucket",
               "ownerIdentity":{
                  "principalId":"A3NL1KOZZKExample"
               },
               "arn":"arn:aws:s3:::mybucket"
            },
            "object":{
```

```
                "key":"HappyFace.jpg",
                "size":1024,
                "eTag":"d41d8cd98f00b204e9800998ecf8427e",
                "versionId":"096fKKXTRTtl3on89fVO.nfljtsv6qko",
                "sequencer":"0055AED6DCD90281E5"
            }
        }
    }
  ]
}
```

Per la definizione di ciascun prefisso di identificazione IAM (ad esempio AIDA, AROA, AGPA), consulta Identificatori IAM nella Guida per l'utente di IAM.

Usando EventBridge

Amazon S3 può inviare eventi ad Amazon EventBridge ogni volta che si verificano determinati eventi nel tuo bucket. A differenza di altre destinazioni, non è necessario selezionare i tipi di eventi che si desidera inviare. Dopo averlo EventBridge abilitato, tutti gli eventi seguenti vengono inviati a. EventBridge È possibile utilizzare EventBridge le regole per indirizzare gli eventi verso destinazioni aggiuntive. Di seguito sono elencati gli eventi a cui invia Amazon S3. EventBridge

Tipo di evento	Descrizione
Oggetto creato	Un oggetto è stato creato. Il campo reason nella struttura del messaggio dell'evento indica quale API S3 è stata utilizzata per creare l'oggetto: PutObject, POST Object o. CopyObjectCompleteMultipartUpload
Oggetto eliminato () DeleteObject Oggetto eliminato (scadenza del ciclo di vita)	Un oggetto è stato eliminato. Quando un oggetto viene eliminato utilizzando una chiamata API S3, il campo del motivo è impostato DeleteObject su. Quando un oggetto viene eliminato da una regola di scadenza del ciclo di vita S3, il campo motivo è impostato su Scadenza ciclo di vita. Per ulteriori informazioni, consulta Oggetti in scadenza.

Tipo di evento	Descrizione
	Quando viene eliminato un oggetto senza versione o quando viene eliminato in modo permanente un oggetto con versione, il campo del tipo di eliminazione viene impostato su Eliminato definitivamente. Quando viene creato un contrassegno di eliminazione per un oggetto con versione, il campo del tipo di eliminazione viene impostato su Contrassegno di eliminazione creato. Per ulteriori informazioni, consulta Eliminazione di versioni di oggetti da un bucket con funzione Controllo delle versioni abilitata.
Ripristino oggetti avviato	È stato avviato un ripristino degli oggetti dalla classe di archiviazione S3 Glacier o S3 Glacier Deep Archive oppure dal livello S3 Intelligent-Tiering Archive Access o Deep Archive Access. Per ulteriori informazioni, consulta Utilizzo di oggetti archiviati.
Ripristino oggetti completato	È stato completato un ripristino di oggetti.
Ripristino oggetti scaduto	La copia temporanea di un oggetto ripristinato da S3 Glacier o S3 Glacier Deep Archive è scaduta ed è stata eliminata.
Classe di archiviazione di oggetti modificata	Un oggetto è stato trasferito a una classe di archiviazione diversa. Per ulteriori informazioni, consulta Trasferimento degli oggetti utilizzando il ciclo di vita Amazon S3.
Livello di accesso agli oggetti modificato	Un oggetto è stato trasferito al livello S3 Intelligent-Tierin g Archive Access o Deep Archive Access. Per ulteriori informazioni, consulta Amazon S3 Intelligent-Tiering.
Aggiornamento dell'ACL dell'ogge tto	L'elenco di controllo degli accessi (ACL) di un oggetto è stato impostato utilizzando PutObject ACL. Un evento non viene generato quando una richiesta non comporta alcuna modifica all'ACL di un oggetto. Per ulteriori informazioni, consulta Panoramica delle liste di controllo accessi (ACL).

Tipo di evento	Descrizione
Aggiunti tag degli oggetti	Un set di tag è stato aggiunto a un oggetto utilizzando. PutObjectTagging Per ulteriori informazioni, consulta Suddivisione in categorie dello storage utilizzando i tag.
Eliminazione di tag degli oggetti	Tutti i tag sono stati rimossi da un oggetto utilizzando DeleteObjectTagging. Per ulteriori informazioni, consulta Suddivisione in categorie dello storage utilizzando i tag.

> **Note**
>
> Per ulteriori informazioni sulla mappatura dei tipi di eventi di Amazon S3 ai tipi di EventBridge evento, consulta. EventBridge Mappatura e risoluzione dei problemi di Amazon

Puoi utilizzare Amazon S3 Event Notifications con EventBridge per scrivere regole che agiscano quando si verifica un evento nel tuo bucket. Ad esempio, è possibile scegliere di ricevere una notifica. Per ulteriori informazioni, consulta Cosa c'è EventBridge nella Amazon EventBridge User Guide.

Per ulteriori informazioni sulle azioni e sui tipi di dati con cui puoi interagire utilizzando l' EventBridge API, consulta l'Amazon EventBridge API Reference nell'Amazon EventBridge API Reference.

Per informazioni sui prezzi, consulta la pagina EventBridge dei prezzi di Amazon.

Argomenti

- EventBridge Autorizzazioni Amazon
- Abilitare Amazon EventBridge
- EventBridge struttura dei messaggi di evento
- EventBridge Mappatura e risoluzione dei problemi di Amazon

EventBridge Autorizzazioni Amazon

Amazon S3 non richiede autorizzazioni aggiuntive per fornire eventi ad Amazon. EventBridge

Abilitare Amazon EventBridge

Puoi abilitare Amazon EventBridge utilizzando la console S3, AWS Command Line Interface (AWS CLI) o l'API REST di Amazon S3.

Utilizzo della console S3

Per abilitare la consegna EventBridge degli eventi nella console S3.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket (Bucket) scegliere il nome del bucket per il quale abilitare gli eventi.
3. Scegliere Properties (Proprietà).
4. Vai alla sezione Notifiche eventi e trova la EventBridge sottosezione Amazon. Scegli Modifica.
5. In Invia notifiche ad Amazon EventBridge per tutti gli eventi in questo bucket scegli Attiva.

> **Note**
>
> Dopo l'attivazione EventBridge, occorrono circa cinque minuti prima che le modifiche abbiano effetto.

Utilizzando il AWS CLI

L'esempio seguente crea una configurazione di notifica bucket per bucket con `DOC-EXAMPLE-BUCKET1` Amazon EventBridge abilitato.

```
aws s3api put-bucket-notification-configuration --bucket example-s3-bucket1 --
notification-configuration='{ "EventBridgeConfiguration": {} }'
```

Utilizzo di REST API

Puoi abilitare Amazon EventBridge su un bucket a livello di codice chiamando l'API REST di Amazon S3. Per ulteriori informazioni, PutBucketNotificationConfigurationconsulta la sezione Amazon Simple Storage Service API Reference.

L'esempio seguente mostra l'XML utilizzato per creare una configurazione di notifica bucket con Amazon EventBridge abilitato.

```
<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <EventBridgeConfiguration>
  </EventBridgeConfiguration>
</NotificationConfiguration>
```

Creazione di regole EventBridge

Una volta abilitato, puoi creare EventBridge regole Amazon per determinate attività. Ad esempio, è possibile inviare notifiche via e-mail quando viene creato un oggetto. Per un tutorial completo, consulta Tutorial: Inviare una notifica quando viene creato un oggetto Amazon S3 nella Amazon EventBridge User Guide.

EventBridge struttura dei messaggi di evento

Il messaggio di notifica inviato da Amazon S3 per pubblicare un evento è in formato JSON. Quando Amazon S3 invia un evento ad Amazon EventBridge, sono presenti i seguenti campi.

- versione — Attualmente 0 (zero) per tutti gli eventi.
- id — Un UUID di quarta versione generato per ogni evento.
- tipo di dettaglio — Il tipo di evento inviato. Per un elenco dei tipi di evento, consulta Usando EventBridge.
- origine — Identifica il servizio che ha originato l'evento.
- account — L'ID a 12 cifre dell' Account AWS del proprietario del bucket.
- time (Ora): momento in cui si è verificato l'evento.
- regione — Identifica la Regione AWS del bucket.
- resources (Risorse): array JSON contenente il nome della risorsa Amazon (ARN) del bucket.
- dettagli— Un oggetto JSON che contiene informazioni sull'evento. Per ulteriori informazioni su ciò che può essere incluso in questo campo, consulta Campo dei dettagli del messaggio di evento.

Esempi di struttura dei messaggi di evento

Di seguito sono riportati alcuni esempi di alcuni messaggi di notifica degli eventi di Amazon S3 che possono essere inviati ad Amazon. EventBridge

Oggetto creato

```
{
```

```
  "version": "0",
  "id": "17793124-05d4-b198-2fde-7ededc63b103",
  "detail-type": "Object Created",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "DOC-EXAMPLE-BUCKET1"
    },
    "object": {
      "key": "example-key",
      "size": 5,
      "etag": "b1946ac92492d2347c6235b4d2611184",
      "version-id": "IYV3p45BT0ac8hjHg1houSdS1a.Mro8e",
      "sequencer": "617f08299329d189"
    },
    "request-id": "N4N7GDK58NMKJ12R",
    "requester": "123456789012",
    "source-ip-address": "1.2.3.4",
    "reason": "PutObject"
  }
}
```

Oggetto eliminato (utilizzando DeleteObject)

```
{
  "version": "0",
  "id": "2ee9cc15-d022-99ea-1fb8-1b1bac4850f9",
  "detail-type": "Object Deleted",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
  ],
```

```
  "detail": {
    "version": "0",
    "bucket": {
      "name": "DOC-EXAMPLE-BUCKET1"
    },
    "object": {
      "key": "example-key",
      "etag": "d41d8cd98f00b204e9800998ecf8427e",
      "version-id": "1QW9g1Z99LUNbvaaYVpW9xDlOLU.qxgF",
      "sequencer": "617f0837b476e463"
    },
    "request-id": "0BH729840619AG5K",
    "requester": "123456789012",
    "source-ip-address": "1.2.3.4",
    "reason": "DeleteObject",
    "deletion-type": "Delete Marker Created"
  }
}
```

Oggetto eliminato (utilizzando la scadenza del ciclo di vita)

```
{
  "version": "0",
  "id": "ad1de317-e409-eba2-9552-30113f8d88e3",
  "detail-type": "Object Deleted",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "DOC-EXAMPLE-BUCKET1"
    },
    "object": {
      "key": "example-key",
      "etag": "d41d8cd98f00b204e9800998ecf8427e",
      "version-id": "mtB0cV.jejK63XkRNceanNMC.qXPWLeK",
      "sequencer": "617b398000000000"
```

```
    },
    "request-id": "20EB74C14654DC47",
    "requester": "s3.amazonaws.com",
    "reason": "Lifecycle Expiration",
    "deletion-type": "Delete Marker Created"
  }
}
```

Ripristino oggetti completato

```
{
  "version": "0",
  "id": "6924de0d-13e2-6bbf-c0c1-b903b753565e",
  "detail-type": "Object Restore Completed",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "DOC-EXAMPLE-BUCKET1"
    },
    "object": {
      "key": "example-key",
      "size": 5,
      "etag": "b1946ac92492d2347c6235b4d2611184",
      "version-id": "KKsjUC1.6gIjqtvhfg5AdMI0eCePIiT3"
    },
    "request-id": "189F19CB7FB1B6A4",
    "requester": "s3.amazonaws.com",
    "restore-expiry-time": "2021-11-13T00:00:00Z",
    "source-storage-class": "GLACIER"
  }
}
```

Campo dei dettagli del messaggio di evento

Il campo dei dettagli contiene un oggetto JSON con informazioni sull'evento. I seguenti campi possono essere presenti nel campo dettagli.

- versione — Attualmente 0 (zero) per tutti gli eventi.
- bucket — Informazioni sul bucket Amazon S3 coinvolto nell'evento.
- oggetto — Informazioni sull'oggetto Amazon S3 coinvolto nell'evento.
- richiesta id — ID della richiesta nella risposta S3.
- richiedente: Account AWS ID o principale del AWS servizio del richiedente.
- source-ip-address— Indirizzo IP di origine della richiesta S3. Presente solo per eventi attivati da una richiesta S3.
- motivo: per gli eventi Object Created, l'API S3 utilizzata per creare l'oggetto: PutObject, POST Object CopyObject, o. CompleteMultipartUpload Per gli eventi Object Deleted, è impostato su DeleteObjectquando un oggetto viene eliminato da una chiamata API S3 o Lifecycle Exployment quando un oggetto viene eliminato da una regola di scadenza del ciclo di vita S3. Per ulteriori informazioni, consulta Oggetti in scadenza.
- tipo di eliminazione — Per eventi Oggetto eliminato, quando viene eliminato un oggetto senza versione o quando viene eliminato in modo permanente un oggetto con versione, questo è impostato su Eliminato permanentemente. Quando viene creato un contrassegno di eliminazione per un oggetto con versione, verrà impostato su Contrassegno di eliminazione creato. Per ulteriori informazioni, consulta Eliminazione di versioni di oggetti da un bucket con funzione Controllo delle versioni abilitata.
- restore-expiry-time— Per gli eventi Object Restore Completed, l'ora in cui la copia temporanea dell'oggetto verrà eliminata da S3. Per ulteriori informazioni, consulta Utilizzo di oggetti archiviati.
- source-storage-class— Per gli eventi Object Restore Initiated e Object Restore Completed, la classe di archiviazione dell'oggetto da ripristinare. Per ulteriori informazioni, consulta Utilizzo di oggetti archiviati.
- destination-storage-class— Per gli eventi Object Storage Class Changed, la nuova classe di archiviazione dell'oggetto. Per ulteriori informazioni, consulta Trasferimento degli oggetti utilizzando il ciclo di vita Amazon S3.
- destination-access-tier— Per gli eventi Object Access Tier Changed, il nuovo livello di accesso dell'oggetto. Per ulteriori informazioni, consulta Amazon S3 Intelligent-Tiering.

EventBridge Mappatura e risoluzione dei problemi di Amazon

La tabella seguente descrive come i tipi di eventi Amazon S3 vengono mappati ai tipi di eventi Amazon EventBridge .

Tipo di evento S3	Tipo di EventBridge dettaglio Amazon
ObjectCreatedTipo di dettaglio Amazon ----Sep----:put ObjectCreated:Put ----Sep----:Post ObjectCreated:Post ----Sep----:Copia ObjectCreated:CompleteMultipartUpload	Oggetto creato
ObjectRemoved:Copia ----sep----:Elimina ObjectRemoved:DeleteMarkerCreated LifecycleExpiration:Elimina ----sep----:Elimina LifecycleExpiration:DeleteMarkerCreated	Oggetto eliminato
ObjectRestore:Elimina ----sep----:Post	Ripristino oggetti avviato
ObjectRestore:Post ----Sep----:Completato	Ripristino oggetti completato
ObjectRestore:Completato ----Sep----:Elimina	Ripristino oggetti scaduto
LifecycleTransition	Classe di archiviazione di oggetti modificata

Tipo di evento S3	Tipo di EventBridge dettaglio Amazon
IntelligentTiering	Livello di accesso agli oggetti modificato
ObjectTagging:Elimina ----Sep---:Inserisci	Aggiunti tag degli oggetti
ObjectTagging:Put ----sep----:Elimina	Eliminazione di tag degli oggetti
ObjectAcl:Elimina ----Sep----:Inserisci	Aggiornamento dell'ACL dell'oggetto

EventBridge Risoluzione dei problemi Amazon

Per informazioni su come risolvere i problemi EventBridge, consulta Troubleshooting Amazon EventBridge nella Amazon EventBridge User Guide.

Utilizzo di analisi e approfondimenti

Puoi utilizzare analisi e informazioni dettagliate in Amazon S3 per comprendere, analizzare e ottimizzare l'utilizzo dello storage. Per ulteriori informazioni, consulta gli argomenti riportati di seguito.

Argomenti

Analisi di Amazon S3 – Analisi della classe di storage

Lo strumento per l'analisi della classe di storage Amazon S3 Analytics consente di analizzare gli schemi di accesso allo storage per stabilire quando eseguire la transizione dei dati corretti alla classe di storage appropriata. Questa nuova funzionalità di analisi di Amazon S3 osserva gli schemi di accesso ai dati per determinare quando è opportuno spostare i dati meno utilizzati dallo storage STANDARD alla classe di storage STANDARD_IA (ad accesso infrequente). Per ulteriori informazioni sulle classi di storage, consulta Utilizzo delle classi di storage di Amazon S3.

Dopo l'øsservazione degli schemi di accesso poco frequenti a un set di dati filtrati in un certo periodo di tempo da parte dell'analisi della classe di archiviazione, i risultati dell'analisi possono essere utilizzati per migliorare le configurazioni del ciclo di vita. È possibile configurare l'analisi della classe di storage per tutti gli oggetti di un bucket, oppure si possono configurare filtri per raggruppare gli oggetti per l'analisi in base a un prefisso condiviso (ossia, oggetti i cui nomi iniziano con una stringa comune), ai tag dell'oggetto o a entrambe le opzioni. Con ogni probabilità, l'applicazione di filtri in base ai gruppi di oggetti si rivelerà la soluzione più vantaggiosa per l'analisi della classe di archiviazione.

> **Important**
>
> L'analisi della classe di archiviazione fornisce solo suggerimenti per le classi da standard a standard (accesso infrequente).

La funzionalità di analisi della classe di archiviazione consente di attivare più filtri per bucket (fino a 1.000) e di ricevere un'analisi separata per ogni filtro. Le configurazioni a più filtri permettono

di analizzare gruppi specifici di oggetti al fine di migliorare le configurazioni del ciclo di vita che trasferiscono gli oggetti alla classe STANDARD_IA.

L'analisi della classe di storage fornisce nella console Amazon S3 le visualizzazioni relative l'utilizzo dello storage aggiornate quotidianamente. Puoi anche esportare questi dati di utilizzo giornaliero in un bucket S3 e visualizzarli in un'applicazione per fogli di calcolo o con strumenti di business intelligence, come Amazon. QuickSight

Ci sono costi associati all'analisi della classe di archiviazione. Per informazioni sui prezzi, consulta Gestione e replica Prezzi di Amazon S3.

Argomenti

Come impostare l'analisi della classe di storage

È possibile impostare l'analisi della classe di storage configurando i dati degli oggetti da analizzare. A seconda della configurazione, l'analisi della classe di storage può:

- Analizzare l'intero contenuto di un bucket.

  Verrà generata un'analisi per tutti gli oggetti del bucket.
- Analizzare gli oggetti raggruppati in base al prefisso e ai tag.

  Si possono configurare filtri che raggruppano gli oggetti per l'analisi in base a un prefisso, ai tag dell'oggetto o a una combinazione di entrambe le opzioni. Viene generata un'analisi separata per ogni filtro configurato. È possibile definire più configurazioni di filtri per ciascun bucket (fino a 1.000).
- Esportare i dati dell'analisi.

  Quando si configura l'analisi della classe di storage per un bucket o un filtro, si può scegliere di far esportare ogni giorno i relativi dati in un file. L'analisi del giorno corrente viene aggiunta al file per formare un registro storico dell'analisi per il filtro configurato. Il file viene aggiornato quotidianamente nella destinazione selezionata. Al momento della selezione dei dati da esportare, si specifica un bucket di destinazione e un prefisso di destinazione (facoltativo) dove scrivere il file.

Puoi utilizzare la console Amazon S3, l'API REST o gli AWS SDK per configurare l' AWS CLI analisi delle classi di storage.

- Per informazioni su come configurare l'analisi delle classi di storage nella console Amazon S3, consulta Configurazione dell'analisi della classe di storage.
- Per utilizzare l'API Amazon S3, usa l'API PutBucketAnalyticsConfigurationREST, o l'equivalente, dagli AWS CLI o AWS SDK.

Come utilizzare l'analisi della classe di storage

L'analisi della classe di storage consente di osservare gli schemi di accesso ai dati nel tempo e raccogliere così informazioni utili per migliorare la gestione del ciclo di vita dello storage STANDARD_IA. Dopo aver configurato un filtro, entro 24 o 48 ore nella console Amazon S3 inizierà a essere disponibile l'analisi dei dati basata sul filtro. Tuttavia, prima di fornire un risultato, la funzionalità continuerà a osservare gli schemi di accesso a un insieme di dati filtrati per almeno 30 giorni, al fine di raccogliere le informazioni necessarie. Dopo i primi risultati, l'analisi resta in esecuzione e aggiorna il risultato via via che gli schemi di accesso cambiano.

Quando configuri un filtro per la prima volta, alla console Amazon S3 potrebbe occorrere qualche istante per analizzare i tuoi dati.

L'analisi della classe di archiviazione continua a osservare gli schemi di accesso a un insieme di dati di oggetti filtrati per almeno 30 giorni, al fine di raccogliere informazioni sufficienti. Una volta raccolte informazioni a sufficienza, nella console Amazon S3 verrà visualizzato un messaggio di analisi completata.

Durante l'analisi degli oggetti ad accesso infrequente, l'analisi della classe di storage prende in considerazione un insieme filtrato di oggetti raggruppati in base al tempo trascorso dal loro caricamento su Amazon S3. e determina se l'accesso al gruppo di età è infrequente valutando i fattori seguenti per l'insieme di dati filtrati:

- Oggetti nella classe di storage STANDARD più grandi di 128 KB.
- Volume medio di storage totale per gruppo di età
- Numero medio di byte trasferiti in uscita (non la frequenza) per gruppo di età.
- I dati dell'esportazione analitica includono solo le richieste con dati pertinenti per l'analisi della classe di storage. Per questo motivo il numero di richieste, nonché il totale dei byte per caricamenti e richieste, potrebbe variare rispetto a quanto riportato nei parametri dello storage o tracciato dai sistemi interni dell'utente

- Benché non siano conteggiate nell'analisi, le richieste GET e PUT non andate a buon fine sono incluse nei parametri dello storage.

Volume di storage recuperato

I grafici della console Amazon S3 mostrano il volume di storage dei dati filtrati che è stato recuperato nel periodo di osservazione.

Percentuale di storage recuperata

I grafici della console Amazon S3 mostrano anche la percentuale di storage dei dati filtrati che è stata recuperata nel periodo di osservazione.

Come indicato più in alto in questo argomento, durante l'analisi degli oggetti ad accesso infrequente, l'analisi della classe di storage prende in considerazione un insieme filtrato di oggetti raggruppati in base al tempo trascorso dal loro caricamento su Amazon S3. L'analisi della classe di storage utilizza questi gruppi predefiniti di età degli oggetti:

- Oggetti Amazon S3 con meno di 15 giorni
- Oggetti Amazon S3 con 15-29 giorni
- Oggetti Amazon S3 con 30-44 giorni
- Oggetti Amazon S3 con 45-59 giorni
- Oggetti Amazon S3 con 60-74 giorni
- Oggetti Amazon S3 con 75-89 giorni
- Oggetti Amazon S3 con 90-119 giorni
- Oggetti Amazon S3 con 120-149 giorni
- Oggetti Amazon S3 con 150-179 giorni
- Oggetti Amazon S3 con 180-364 giorni
- Oggetti Amazon S3 con 365-729 giorni
- Oggetti Amazon S3 con almeno 730 giorni

In genere servono circa 30 giorni di osservazione degli schemi di accesso per raccogliere informazioni sufficienti a generare un risultato dell'analisi. Potrebbero anche essere necessari più di 30 giorni, a seconda dei singoli schemi di accesso ai dati. Tuttavia, dopo aver configurato un filtro, entro 24 o 48 ore nella console Amazon S3 inizierà a essere disponibile l'analisi dei dati basata sul

filtro. Nella console Amazon S3 è possibile visualizzare l'analisi giornaliera dell'accesso agli oggetti suddivisa per gruppo di età degli oggetti.

Volume di storage ad accesso infrequente

La console Amazon S3 mostra i modelli di accesso raggruppati in base ai gruppi di età degli oggetti predefiniti. Il testo visualizzato Frequently accessed (Accesso frequente) o Infrequently accessed (Accesso poco frequente) è inteso come un aiuto visivo per aiutarti nel processo di creazione del ciclo di vita.

Come esportare i dati relativi all'analisi della classe di storage

Si può scegliere di far esportare i report dell'analisi della classe di storage in un file flat in formato CSV. I report sono aggiornati quotidianamente e si basano sui filtri configurati per i gruppi di età degli oggetti. Utilizzando la console Amazon S3, al momento di creare un filtro si può scegliere l'opzione di esportazione del report. Al momento della selezione dei dati da esportare, si specifica un bucket di destinazione e un prefisso di destinazione (facoltativo) dove scrivere il file. I dati possono essere esportati in un bucket di destinazione in un account diverso. ma il bucket di destinazione deve trovarsi nella stessa regione del bucket configurato per l'analisi.

È necessario creare una policy sul bucket di destinazione per concedere le autorizzazioni ad Amazon S3 per verificare a chi Account AWS appartiene il bucket e scrivere oggetti nel bucket nella posizione definita. Per un esempio di policy, consulta Concedere autorizzazioni per S3 Inventory e S3 Analytics.

Una volta configurati, i report dell'analisi della classe di storage esportati iniziano a essere disponibili dopo 24 ore. Successivamente Amazon S3 continua a monitorare e generare esportazioni quotidiane.

Puoi aprire il file CSV in un'applicazione per fogli di calcolo o importare il file in altre applicazioni come Amazon. QuickSight Per informazioni sull'uso dei file Amazon S3 con Amazon QuickSight, consulta Create a Data Set Using Amazon S3 Files nella Amazon User Guide. QuickSight

Nel file esportato, i dati sono ordinati per data nell'ambito del gruppo di età degli oggetti, come nell'esempio seguente. Se la classe di storage è STANDARD, la riga contiene anche i dati per le colonne `ObjectAgeForSIATransition` e `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		

Alla fine del report, il gruppo di età degli oggetti restituito è ALL (Tutti). Le righe ALL contengono i totali cumulativi, inclusi gli oggetti inferiori a 128 KB, per tutti i gruppi di età per quel dato giorno.

8/24/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599		0	000-014
9/3/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599		0.02426125	015-029
8/28/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599		0.03545875	015-029
8/17/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599		0	000-014
8/25/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599		0	000-014
9/6/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599		0.0209529	015-029
9/4/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599		0.02304819	015-029
8/22/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599		0	000-014
8/21/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599		0	000-014
8/30/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599		0.03073092	015-029
8/20/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599		0	000-014

La sezione successiva descrive le colonne utilizzate nel report.

Layout del file esportato

Nella seguente tabella è descritto il layout del file esportato.

Configurazione dell'analisi della classe di storage

Lo strumento per l'analisi della classe di storage Amazon S3 consente di analizzare gli schemi di accesso allo storage per stabilire quando eseguire la transizione dei dati verso l'opportuna classe di storage. Osservando gli schemi di accesso ai dati, l'analisi della classe di storage aiuta a determinare quando è opportuno spostare i dati meno utilizzati dallo storage STANDARD alla classe di storage STANDARD_IA (ad accesso infrequente). Per maggiori informazioni su STANDARD_IA, consulta le domande frequenti su Amazon S3 e Utilizzo delle classi di storage di Amazon S3.

È possibile impostare l'analisi della classe di storage configurando i dati degli oggetti da analizzare. A seconda della configurazione, l'analisi della classe di storage può:

- Analizzare l'intero contenuto di un bucket.

  Verrà generata un'analisi per tutti gli oggetti del bucket.
- Analizzare gli oggetti raggruppati in base al prefisso e ai tag.

  Si possono configurare filtri che raggruppano gli oggetti per l'analisi in base a un prefisso, ai tag dell'oggetto o a una combinazione di entrambe le opzioni. Viene generata un'analisi separata per ogni filtro configurato. È possibile definire più configurazioni di filtri per ciascun bucket (fino a 1.000).
- Esportare i dati dell'analisi.

  Quando si configura l'analisi della classe di storage per un bucket o un filtro, si può scegliere di far esportare ogni giorno i relativi dati in un file. L'analisi del giorno corrente viene aggiunta al file per formare un registro storico dell'analisi per il filtro configurato. Il file viene aggiornato

quotidianamente nella destinazione selezionata. Al momento della selezione dei dati da esportare, si specifica un bucket di destinazione e un prefisso di destinazione (facoltativo) dove scrivere il file.

Puoi utilizzare la console Amazon S3, l'API REST o gli AWS SDK per configurare l' AWS CLI analisi delle classi di storage.

> **Important**
>
> L'analisi della classe di archiviazione non fornisce suggerimenti sulla transizione alle classi di archiviazione ONEZONE_IA o S3 Glacier Flexible Retrieval.
> Se desideri configurare l'analisi della classe di storage per esportare i risultati come file.csv e il bucket di destinazione utilizza la crittografia dei bucket predefinita con a AWS KMS key, devi aggiornare la policy AWS KMS chiave per concedere ad Amazon S3 l'autorizzazione a crittografare il file.csv. Per istruzioni, consulta Concessione ad Amazon S3 dell'autorizzazione per l'utilizzo della chiave gestita dal cliente per la crittografia.

Per ulteriori informazioni sull'analisi, consulta Analisi di Amazon S3 – Analisi della classe di storage.

Utilizzo della console S3

Per configurare l'analisi della classe di storage

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket), scegliere il nome del bucket per cui si desidera configurare l'analisi della classe di storage.
3. Seleziona la scheda Parametri.
4. In Storage Class Analysis (Analisi classe storage), scegliere Create analytics configuration (Crea configurazione di analisi).
5. Digitare un nome per il filtro. Per analizzare l'intero bucket, lasciare vuoto il campo Prefix (Prefisso).
6. Nel campo Prefix (Prefisso), digitare il testo per il prefisso per gli oggetti che si desidera analizzare.
7. Per aggiungere un tag, scegli Add tag (Aggiungi tag). Digitare una Key (Chiave) e un Value (Valore) per il tag. È possibile immettere un prefisso e più tag.

8. Se si desidera, scegliere Enable (Abilita) in Export CSV (Esporta CSV) per esportare i report dell'analisi in un file flat in formato .csv. Scegliere un bucket di destinazione per archiviare il file. È anche possibile scegliere un prefisso per il bucket di destinazione. Il bucket di destinazione deve trovarsi nello Regione AWS stesso bucket per il quale stai configurando l'analisi. Il bucket di destinazione può trovarsi in un diverso Account AWS.

   Se il bucket di destinazione per il file.csv utilizza la crittografia dei bucket predefinita con una chiave KMS, devi aggiornare la policy delle chiavi per concedere ad Amazon S3 l' AWS KMS autorizzazione a crittografare il file.csv. Per istruzioni, consulta Concessione ad Amazon S3 dell'autorizzazione per l'utilizzo della chiave gestita dal cliente per la crittografia.
9. Scegliere Create configuration (Crea configurazione).

Amazon S3 crea una policy nel bucket di destinazione che concede ad Amazon S3 l'autorizzazione di scrittura. Ciò gli consentirà di scrivere i dati di esportazione nel bucket.

Se si verifica un errore quando si tenta di creare la policy di bucket, vengono fornite le istruzioni per correggerlo. Ad esempio, se hai scelto un bucket di destinazione in un altro Account AWS e non disponi delle autorizzazioni di lettura e scrittura per la policy di bucket, verrà visualizzato il messaggio riportato di seguito. Il proprietario del bucket di destinazione deve aggiungere a quest'ultimo la policy di bucket. Se la policy non viene aggiunta al bucket di destinazione, i dati non verranno esportati in quanto Amazon S3 non dispone dell'autorizzazione di scrittura su tale bucket. Se il bucket di origine è di proprietà di un account diverso da quello dell'utente attuale, l'ID account corretto del bucket di origine verrà sostituito nella policy.

Per informazioni sui dati esportati e sul funzionamento del filtro, consulta Analisi di Amazon S3 – Analisi della classe di storage.

Utilizzo di REST API

Per configurare Storage Class Analysis utilizzando l'API REST, usa. PutBucketAnalyticsConfiguration Puoi anche utilizzare l'operazione equivalente con AWS CLI o AWS SDK.

È possibile utilizzare le seguenti API REST per lavorare con l'analisi della classe di storage:

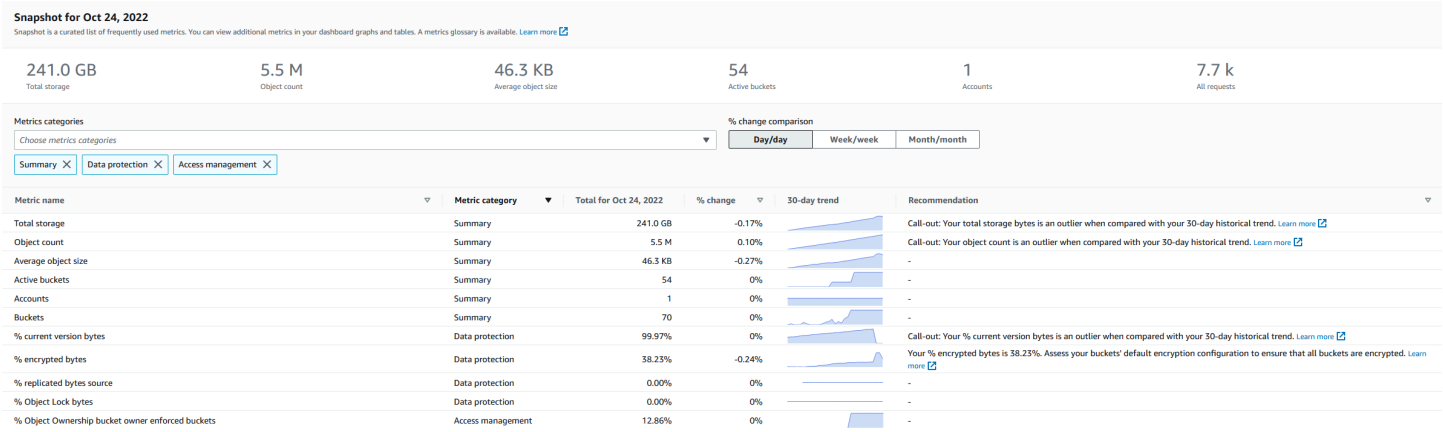
- Configurazione di DELETE Bucket Analytics
- Configurazione di GET Bucket Analytics
- List Bucket Analytics Configuration

Valutazione dell'attività e dell'utilizzo dello storage con Amazon S3 Storage Lens

Amazon S3 Storage Lens è una funzionalità di analisi dell'archiviazione su cloud che permette di avere una panoramica completa a livello di organizzazione sull'utilizzo e sull'archiviazione di oggetti. S3 Storage Lens analizza i parametri di archiviazione per fornire raccomandazioni contestuali che puoi usare per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati.

Puoi utilizzare i parametri di S3 Storage Lens per generare approfondimenti di riepilogo. Ad esempio, per scoprire la quantità di spazio di archiviazione disponibile in tutta l'organizzazione o quali sono i bucket e i prefissi caratterizzati da una crescita più rapida. Puoi utilizzare i parametri di Amazon S3 Storage Lens anche per individuare le opportunità di ottimizzazione dei costi, implementare le best practice di protezione dei dati e gestione degli accessi e migliorare le prestazioni dei carichi di lavoro delle applicazioni. Ad esempio, puoi identificare i bucket che non hanno regole del ciclo di vita S3 per interrompere i caricamenti in più parti incompleti che risalgono a più di 7 giorni. Puoi anche individuare i bucket non conformi alle best practice di protezione dei dati, come quelli che usano la replica S3 o il controllo delle versioni S3.

S3 Storage Lens aggrega i tuoi parametri e mostra le informazioni nella sezione Account snapshot (Snapshot dell'account) nella pagina Buckets (Bucket) della console di Amazon S3. S3 Storage Lens dispone inoltre di un pannello di controllo interattivo che puoi usare per visualizzare informazioni dettagliate e tendenze, contrassegnare le anomalie e ricevere consigli per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati. Nel pannello di controllo sono disponibili opzioni di drill-down per generare e visualizzare approfondimenti a livello di organizzazione, account, Regione AWS, classe di storage, bucket, prefisso o gruppo Storage Lens. Puoi anche inviare un'esportazione dei parametri quotidiana in formato CSV o Parquet a un bucket S3.



Caratteristiche e parametri di S3 Storage Lens

In S3 Storage Lens è disponibile un pannello di controllo interattivo predefinito che viene aggiornato quotidianamente. S3 Storage Lens preconfigura questo pannello di controllo per visualizzare le informazioni dettagliate di riepilogo e le tendenze per l'intero account e le aggiorna quotidianamente nella console S3. I parametri di questo pannello di controllo vengono riepilogati anche nello snapshot dell'account nella paginaBuckets (Bucket). Per ulteriori informazioni, consulta Pannello di controllo predefinito.

Per creare altri pannelli di controllo e definirne l'ambito in base a Regioni AWS, bucket S3 o account (per AWS Organizations), devi creare una configurazione per il pannello di controllo di S3 Storage Lens. Puoi creare e gestire le configurazioni del pannello di controllo di S3 Storage Lens utilizzando la console Amazon S3, la AWS Command Line Interface (AWS CLI), gli SDK AWS o la REST API Amazon S3. Quando crei o modifichi un pannello di controllo di S3 Storage Lens, ne definisci l'ambito e la selezione dei parametri.

A un costo aggiuntivo, potrai eseguire l'aggiornamento e ricevere suggerimenti e parametri avanzati di S3 Storage Lens. I parametri avanzati e i suggerimenti ti consentono di accedere a parametri e funzionalità aggiuntive per ottenere informazioni dettagliate sul tuo spazio di archiviazione. Queste funzionalità includono categorie di parametri avanzati, aggregazione a livello di prefisso, suggerimenti contestuali e funzionalità di pubblicazione Amazon CloudWatch. L'aggregazione a livello di prefisso e i suggerimenti contestuali sono disponibili solo nella console di Amazon S3. Per maggiori informazioni sui prezzi di S3 Storage Lens, consulta i prezzi di Amazon S3.

Categorie di parametri

All'interno dei livelli gratuiti e avanzati, i parametri sono organizzati in categorie in linea con i principali casi d'uso, come l'ottimizzazione dei costi e la protezione dei dati. I parametri gratuiti includono parametri per riepilogo, ottimizzazione dei costi, protezione dei dati, gestione degli accessi, prestazioni ed eventi. Quando esegui l'aggiornamento a parametri e suggerimenti avanzati, puoi abilitare i parametri avanzati relativi a ottimizzazione dei costi e protezione dei dati. Puoi utilizzare questi parametri avanzati per ridurre ulteriormente i costi di archiviazione S3 e migliorare la tua posizione nei confronti della protezione dei dati. Puoi anche abilitare i parametri relativi alle attività e quelli relativi ai codici di stato dettagliati per migliorare le prestazioni dei carichi di lavoro delle applicazioni che accedono ai bucket S3. Per ulteriori informazioni sulle categorie di parametri gratuiti e avanzati, consulta Selezione dei parametri.

Puoi valutare la tua archiviazione in base alle best practice S3, ad esempio per analizzare la percentuale di bucket per i quali è abilitata la crittografia o il blocco degli oggetti S3 o la funzionalità

S3 di controllo delle versioni. È anche possibile individuare potenziali opportunità di risparmio sui costi. Ad esempio, puoi utilizzare i parametri relativi al conteggio delle regole del ciclo di vita S3 per identificare i bucket senza regole di scadenza o di transizione del ciclo di vita. Puoi anche analizzare l'attività di richiesta per bucket per individuare i bucket in cui gli oggetti possono essere trasferiti a una classe di archiviazione con costi più bassi. Per ulteriori informazioni, consulta Casi d'uso relativi ai parametri di Amazon S3 Storage Lens.

Esportazione dei parametri

Oltre al pannello di controllo della console S3, è possibile esportare parametri in formato CSV o Parquet in un bucket S3 per ulteriori analisi mediante lo strumento di analisi che desideri. Per ulteriori informazioni, consulta Visualizzazione dei parametri di Amazon S3 Storage Lens utilizzando una esportazione di dati.

Pubblicazione Amazon CloudWatch

Puoi pubblicare i parametri di utilizzo e attività di S3 Storage Lens su Amazon CloudWatch per creare una visualizzazione unificata dell'integrità operativa nei pannelli di controllo di CloudWatch. È inoltre possibile utilizzare le funzioni di CloudWatch, come allarmi e azioni attivate, matematica dei parametri e rilevamento delle anomalie, per monitorare e intervenire sui parametri di S3 Storage Lens. Inoltre, le operazioni API di CloudWatch consentono alle applicazioni, inclusi i provider di terze parti, di accedere ai parametri di S3 Storage Lens. L'opzione di pubblicazione CloudWatch è disponibile per i pannelli di controllo aggiornati in base all'opzione S3 Storage Lens Advanced metrics and recommendations (Parametri e suggerimenti avanzati). Per ulteriori informazioni sul supporto di parametri per S3 Storage Lens su CloudWatch, vedi Monitoraggio dei parametri di S3 Storage Lens in CloudWatch.

Per ulteriori informazioni sull'utilizzo di S3 Storage Lens, consulta i seguenti argomenti.

Argomenti

- Informazioni su Amazon S3 Storage Lens
- Utilizzo di Amazon S3 Storage Lens con AWS Organizations
- Autorizzazioni Amazon S3 Storage Lens
- Visualizzazione dei parametri con Amazon S3 Storage Lens
- Casi d'uso relativi ai parametri di Amazon S3 Storage Lens
- Glossario dei parametri di Amazon S3 Storage Lens
- Utilizzo di Amazon S3 Storage Lens con la console e l'API

- Utilizzo dei gruppi S3 Storage Lens

Informazioni su Amazon S3 Storage Lens

> **Important**
>
> Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. A partire dal 5 gennaio 2023, tutti i caricamenti di nuovi oggetti su Amazon S3 vengono crittografati automaticamente senza costi aggiuntivi e senza alcun impatto sulle prestazioni. Lo stato della crittografia automatica per la configurazione della crittografia predefinita del bucket S3 e per il caricamento di nuovi oggetti è disponibile nei log di AWS CloudTrail, in S3 Inventory, in S3 Storage Lens, nella console di Amazon S3 e come intestazione di risposta API Amazon S3 aggiuntiva nella AWS Command Line Interface e negli SDK AWS. Per ulteriori informazioni, consulta Domande frequenti sulla crittografia predefinita.

Amazon S3 Storage Lens è una funzionalità di analisi dell'archiviazione su cloud che puoi utilizzare per avere una panoramica completa a livello di organizzazione sull'utilizzo e sulle attività relative all'archiviazione di oggetti. È possibile utilizzare i parametri di S3 Storage Lens per generare approfondimenti, ad esempio per scoprire la quantità di spazio di archiviazione disponibile nell'intera organizzazione o quali sono i bucket e i prefissi caratterizzati da una crescita più rapida. Puoi anche utilizzare i parametri di S3 Storage Lens per individuare le opportunità di ottimizzazione dei costi, implementare le best practice di protezione e sicurezza dei dati e migliorare le prestazioni dei carichi di lavoro delle applicazioni. Ad esempio, puoi identificare i bucket che non hanno regole del ciclo di vita S3 per far scadere i caricamenti in più parti incompleti che risalgono a più di 7 giorni. Puoi anche individuare i bucket non conformi alle best practice di protezione dei dati, come quelli che usano la replica S3 o il controllo delle versioni S3. S3 Storage Lens analizza i parametri di archiviazione per fornire raccomandazioni contestuali che puoi usare per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati.

S3 Storage Lens aggrega i tuoi parametri e mostra le informazioni nella sezione Account snapshot (Snapshot dell'account) nella pagina Buckets (Bucket) della console di Amazon S3. S3 Storage Lens dispone inoltre di un pannello di controllo interattivo che puoi usare per visualizzare informazioni dettagliate e tendenze, contrassegnare le anomalie e ricevere consigli per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati. Nel pannello di controllo sono disponibili opzioni di drill-down per generare e visualizzare approfondimenti a livello di

organizzazione, account, Regione AWS, classe di storage, bucket, prefisso o gruppo Storage Lens. Puoi anche inviare un'esportazione dei parametri quotidiana in formato CSV o Parquet a un bucket S3. Puoi creare e gestire i pannelli di controllo di S3 Storage Lens utilizzando la console Amazon S3, la AWS Command Line Interface (AWS CLI), gli SDK AWS o la REST API Amazon S3.

Concetti e terminologia di S3 Storage Lens

Questa sezione contiene la terminologia e i concetti essenziali per comprendere e utilizzare correttamente Amazon S3 Storage Lens.

Argomenti

Configurazione del pannello di controllo

S3 Storage Lens richiede una configurazione del pannello di controllo contenente le proprietà necessarie per aggregare i parametri per tuo conto per un singolo pannello di controllo o un'esportazione. Quando crei una configurazione, scegli il nome del pannello di controllo e la regione principale, che non puoi modificare successivamente alla creazione del pannello di controllo. Facoltativamente, puoi aggiungere tag e configurare un'esportazione di parametri in formato CSV o Parquet.

Nella configurazione del pannello di controllo, definisci anche l'ambito del pannello e la selezione dei parametri. L'ambito può includere tutto lo spazio di archiviazione per l'account o le sezioni dell'organizzazione filtrati per regione, bucket e account. Quando configuri la selezione dei parametri,

puoi scegliere tra parametri gratuiti e parametri e suggerimenti avanzati, a cui puoi accedere con l'aggiornamento a un costo aggiuntivo. I parametri avanzati e i suggerimenti ti consentono di accedere a parametri e funzionalità aggiuntive. Queste funzionalità includono categorie di parametri avanzati, aggregazione a livello di prefisso, suggerimenti contestuali e funzionalità di pubblicazione Amazon CloudWatch. Per maggiori informazioni sui prezzi di S3 Storage Lens, consulta i prezzi di Amazon S3.

Pannello di controllo predefinito

Il pannello di controllo predefinito di S3 Storage Lens nella console è denominato default-account-dashboard. S3 preconfigura questo pannello di controllo per visualizzare le informazioni dettagliate di riepilogo e le tendenze per l'intero account e le aggiorna quotidianamente nella console S3. Non è possibile modificare l'ambito di configurazione del pannello di controllo predefinito, ma è possibile aggiornare la selezione dei parametri dai parametri gratuiti ai suggerimenti e parametri avanzati a pagamento. Puoi configurare l'esportazione facoltativa dei parametri o addirittura disabilitare il pannello di controllo. Tuttavia, il pannello di controllo predefinito non può essere eliminato.

> **Note**
>
> In caso di disattivazione del pannello di controllo predefinito, non viene più aggiornato. Non riceverai più alcun nuovo parametro giornaliero in S3 Storage Lens, nell'esportazione dei parametri o nello snapshot dell'account nella pagina Bucket S3. Se la dashboard utilizza parametri e suggerimenti avanzati, non ti verrà più addebitato alcun costo. Puoi comunque visualizzare i dati della cronologia nel pannello di controllo fino alla scadenza delle query di dati (14 giorni). Questo periodo è di 15 mesi se hai abilitato i parametri avanzati e i suggerimenti. Per accedere ai dati della cronologia, puoi riattivare il pannello di controllo entro il periodo di scadenza.

Pannelli di controllo

Puoi inoltre creare altri pannelli di controllo di S3 Storage Lens e definirne l'ambito per Regioni AWS, bucket S3 o account (per AWS Organizations). Quando crei o modifichi un pannello di controllo di S3 Storage Lens, ne definisci l'ambito e la selezione dei parametri. A un costo aggiuntivo, potrai eseguire l'aggiornamento e ricevere suggerimenti e parametri avanzati di S3 Storage Lens. I parametri avanzati e i suggerimenti ti consentono di accedere a parametri e funzionalità aggiuntive per ottenere informazioni dettagliate sul tuo spazio di archiviazione. Queste funzionalità includono categorie di parametri avanzati, aggregazione a livello di prefisso, suggerimenti contestuali e funzionalità

di pubblicazione Amazon CloudWatch. Per maggiori informazioni sui prezzi di S3 Storage Lens, consulta i prezzi di Amazon S3.

Puoi anche disabilitare o eliminare i pannelli di controllo. Se disattivi un pannello di controllo, questo non sarà più aggiornato e non riceverai più nuovi parametri giornalieri. Puoi comunque visualizzare i dati della cronologia fino al periodo di scadenza di 14 giorni. Se hai abilitato i parametri avanzati e i suggerimenti per il pannello di controllo, questo periodo è di 15 mesi. Per accedere ai dati della cronologia, puoi riattivare il pannello di controllo entro il periodo di scadenza.

Se si elimina il pannello di controllo, tutte le impostazioni di configurazione del pannello di controllo saranno perse. Non riceverai più nuovi parametri giornalieri e perderai anche l'accesso ai dati della cronologia associati a tale pannello di controllo. Se desideri accedere ai dati della cronologia di un pannello di controllo eliminato, dovrai creare un altro pannello di controllo con lo stesso nome nella stessa regione di origine.

> **Note**
>
> - S3 Storage Lens può essere utilizzato per creare fino a 50 pannelli di controllo per ogni regione.
> - I pannelli di controllo a livello di organizzazione possono essere limitati solo a un ambito regionale.

Snapshot dell'account

Lo snapshot dell'account S3 Storage Lens riepiloga i parametri del pannello di controllo predefinito e mostra l'archiviazione totale, il numero di oggetti e la dimensione media degli oggetti nella pagina Buckets (Bucket) della console S3. Questo snapshot dell'account consente di accedere rapidamente a informazioni dettagliate sullo spazio di archiviazione senza dover uscire dalla pagina Buckets (Bucket). Lo snapshot dell'account fornisce anche l'accesso con un clic al pannello di controllo interattivo di S3 Storage Lens.

È possibile utilizzare il pannello di controllo per visualizzare informazioni dettagliate e tendenze, contrassegnare le anomalie e ricevere suggerimenti per ottimizzare i costi di archiviazione e applicare best practice per la protezione dei dati. Nel pannello di controllo sono disponibili opzioni di drill-down per generare informazioni dettagliate a livello di organizzazione, account, bucket, oggetto o prefisso. Puoi anche inviare un'esportazione di parametri una volta al giorno a un bucket S3 in formato CSV o Parquet.

Non è possibile modificare l'ambito del pannello di controllo dell'account predefinito perché è collegato allo snapshot dell'account. Puoi tuttavia aggiornare la selezione dei parametri nel pannello di controllo dell'account predefinito da parametri gratuiti a parametri e suggerimenti avanzati a pagamento. Dopo l'aggiornamento, è quindi possibile visualizzare tutte le richieste, i byte caricati e i byte scaricati nello snapshot dell'account S3 Storage Lens.

> **Note**
>
> In caso di disattivazione del pannello di controllo predefinito, lo snapshot dell'account non viene più aggiornata. Per riprendere la visualizzazione dei parametri nello snapshot dell'account, è possibile riattivare il pannello di controllo dell'account predefinito.

Esportazione dei parametri

Una esportazione dei parametri di S3 Storage Lens è un file che contiene tutti i parametri identificati nella configurazione di S3 Storage Lens. Queste informazioni vengono generate quotidianamente in formato CSV o Parquet e vengono inviate a un bucket S3. Puoi utilizzare l'esportazione dei parametri metriche per ulteriori analisi utilizzando lo strumento per i parametri di tua scelta. Il bucket S3 per l'esportazione dei parametri deve trovarsi nella stessa regione della configurazione di S3 Storage Lens. Puoi generare un'esportazione dei parametri di S3 Storage Lens dalla console S3 modificando la configurazione del pannello di controllo. Puoi anche configurare un'esportazione di parametri utilizzando la AWS CLI e gli SDK AWS.

Regione di origine

La regione di origine è la Regione AWS in cui vengono memorizzati tutti i parametri di S3 Storage Lens per una determinata configurazione del pannello di controllo. Quando crei la configurazione del pannello di controllo di S3 Storage Lens, dovrai scegliere una regione di origine. Dopo aver scelto una regione di origine, non puoi più cambiarla. Inoltre, se crei un gruppo Storage Lens, ti consigliamo di scegliere la stessa regione del pannello di controllo di Storage Lens.

> **Note**
>
> Come regione di origine puoi scegliere una delle seguenti regioni:
>
> - Stati Uniti orientali (Virginia settentrionale) – `us-east-1`
> - Stati Uniti orientali (Ohio) – `us-east-2`
> - Stati Uniti occidentali (California settentrionale) – `us-west-1`

- Stati Uniti occidentali (Oregon) – `us-west-2`
- Asia Pacifico (Mumbai) – `ap-south-1`
- Asia Pacifico (Seul) - `ap-northeast-2`
- Asia Pacifico (Singapore) – `ap-southeast-1`
- Asia Pacifico (Sydney) - `ap-southeast-2`
- Asia Pacifico (Tokyo) - `ap-northeast-1`
- Canada (Central) – `ca-central-1`
- Cina (Pechino): `cn-north-1`
- Cina (Ningxia): `cn-northwest-1`
- Europe (Francoforte) – `eu-central-1`
- Europa (Irlanda) – `eu-west-1`
- Europe (Londra) – `eu-west-2`
- Europe (Paris) – `eu-west-3`
- Europe (Stockholm) – `eu-north-1`
- Sud America (San Paolo) – `sa-east-1`

Periodo di conservazione

I parametri di S3 Storage Lens vengono conservati in modo da poter vedere le tendenze cronologiche e confrontare le differenze in termini di archiviazione e attività nel tempo. I parametri di Amazon S3 Storage Lens per le query possono essere utilizzate in modo da poter vedere le tendenze cronologiche e confrontare le differenze nell'utilizzo e nell'attività di archiviazione nel tempo.

Tutti i parametri S3 Storage Lens sono conservati per un periodo di 15 mesi. Tuttavia, i parametri sono disponibili solo per le query per una durata specifica, che dipende dalla selezione dei parametri. Questa durata non può essere modificata. Sono disponibili parametri gratuiti per query per un periodo di 14 giorni, mentre quelli avanzati per query per 15 mesi.

Categorie di parametri

All'interno dei livelli gratuiti e avanzati, i parametri di S3 Storage Lens sono organizzati in categorie in linea con i principali casi d'uso, come l'ottimizzazione dei costi e la protezione dei dati. I parametri gratuiti includono parametri per riepilogo, ottimizzazione dei costi, protezione dei dati, gestione degli accessi, prestazioni ed eventi. Quando esegui l'aggiornamento a parametri e suggerimenti avanzati,

puoi abilitare ulteriori parametri di ottimizzazione dei costi e protezione dei dati che puoi utilizzare per ridurre ulteriormente i costi di archiviazione S3 e garantire la protezione dei dati. Puoi anche abilitare i parametri delle attività e i parametri dei codici di stato dettagliati che puoi utilizzare per migliorare le prestazioni dei flussi di lavoro delle applicazioni.

L'elenco seguente mostra tutte le categorie di parametri gratuiti e avanzati. Per un elenco completo dei singoli parametri inclusi in ciascuna categoria, consulta la sezione Glossario dei parametri.

Parametri di riepilogo

I parametri di riepilogo forniscono informazioni generali sull'archiviazione S3, inclusi i byte totali di archiviazione e il conteggio degli oggetti.

Parametri per l'ottimizzazione dei costi

I parametri per l'ottimizzazione dei costi forniscono informazioni che puoi utilizzare per gestire e ottimizzare i costi di archiviazione. Ad esempio, puoi identificare i bucket con caricamenti in più parti incompleti che risalgono a più di 7 giorni fa.

Con i parametri avanzati e i suggerimenti, puoi abilitare i parametri avanzati per l'ottimizzazione dei costi. Questi parametri includono i parametri relativi al conteggio delle regole del ciclo di vita S3 che puoi utilizzare per ottenere i conteggi delle regole del ciclo di vita S3 per ogni bucket.

Parametri per la protezione dei dati

I parametri per la protezione dei dati forniscono informazioni sulle funzionalità di protezione dei dati, come la crittografia e il controllo delle versioni S3. Puoi utilizzare questi parametri per identificare i bucket non conformi alle best practice per la protezione dei dati. Ad esempio, è possibile identificare i bucket che non utilizzano la crittografia predefinita con chiavi AWS Key Management Service (SSE-KMS) o la funzionalità S3 di controllo delle versioni.

Con i parametri avanzati e i suggerimenti, puoi abilitare i parametri avanzati per la protezione dei dati. Questi parametri includono i parametri relativi al conteggio delle regole di replica per bucket.

Parametri per la gestione degli accessi

I parametri per la gestione degli accessi forniscono informazioni sulla caratteristica S3 Object Ownership. Puoi utilizzare questi parametri per visualizzare le impostazioni di Object Ownership usate dai tuoi bucket.

Parametri degli eventi

I parametri degli eventi forniscono approfondimenti relativi alla funzionalità S3 di notifica eventi. Con i parametri degli eventi, puoi vedere in quali bucket è configurata la funzionalità S3 di notifica eventi.

Parametri prestazionali

I parametri relativi alle prestazioni forniscono informazioni su Accelerazione del trasferimento Amazon S3 (Amazon S3TA). Con i parametri relativi alle prestazioni, puoi vedere in quali bucket è abilitata la funzionalità di accelerazione del trasferimento.

Parametri delle attività (avanzati)

Se aggiorni il pannello di controllo a Raccomandazioni e parametri avanzati, puoi abilitare i parametri relativi delle attività. I parametri delle attività mostrano dettagli sulla modalità con cui viene richiesto lo spazio di archiviazione, ad esempio tutte le richieste, richieste Get, richieste Put, byte caricati o scaricati ed errori.

Le metriche di attività a livello di prefisso possono aiutarti a determinare quali prefissi vengono utilizzati di rado, in modo da poter passare a una classe di archiviazione ottimale utilizzando S3 Lifecycle.

Parametri dei codici di stato dettagliati (avanzati)

Se aggiorni il pannello di controllo a Raccomandazioni e parametri avanzati, puoi abilitare i parametri relativi ai codici di stato dettagliati. I parametri relativi ai codici di stato dettagliati forniscono informazioni sui codici di stato HTTP, come 403 Accesso negato e 503 Servizio non disponibile, che puoi utilizzare per risolvere problemi di accesso o prestazioni. Ad esempio, puoi esaminare il parametro 403 Forbidden error count (Conteggio errori 403 Accesso negato) per identificare i carichi di lavoro che accedono ai bucket senza le autorizzazioni corrette applicate.

Utilizzando i parametri relativi ai codici di stato dettagliati a livello di prefisso puoi comprendere meglio le occorrenze del codice di stato HTTP per prefisso. Ad esempio, le metriche del conteggio degli errori 503 consentono di identificare i prefissi che ricevono richieste di limitazione durante l'importazione dei dati.

Raccomandazioni

S3 Storage Lens fornisce raccomandazioni automatizzate per ottimizzare lo storage. Le raccomandazioni vengono posizionate contestualmente insieme ai parametri pertinenti nel pannello di controllo di S3 Storage Lens. I dati della cronologia non sono idonei per le raccomandazioni

in quanto le raccomandazioni sono rilevanti per quanto sta accadendo nel periodo più recente. I suggerimenti appaiono solo quando sono rilevanti.

Le raccomandazioni di S3 Storage Lens sono disponibili nelle seguenti forme:

- Suggerimenti

  I suggerimenti segnalano le tendenze all'interno dell'archiviazione e delle attività che potrebbero indicare un'opportunità di ottimizzazione dei costi di archiviazione o fare riferimento a una best practice per la protezione dei dati. Per maggiori dettagli sulle regioni, i bucket o i prefissi specifici, consulta gli argomenti riportati nella Guida per l'utente di Amazon S3 e nel pannello di controllo di S3 Storage Lens.
- Callout

  I callout sono raccomandazioni che avvisano l'utente riguardo ad anomalie interessanti in termini di archiviazione nell'arco di un periodo che potrebbero richiedere una maggiore attenzione o monitoraggio.
  - Callout anomalie

    S3 Storage Lens fornisce callout per i parametri che sono valori anomali, in base alla recente tendenza di 30 giorni. L'anomalia viene calcolata utilizzando un punteggio standard, noto anche come z-score. Per ottenere questo punteggio, il parametro del giorno corrente viene sottratto dalla media dei valori del parametro degli ultimi 30 giorni. Il valore del parametro del giorno corrente viene quindi diviso per la deviazione standard di tale parametro negli ultimi 30 giorni. Il punteggio risultante è solitamente compreso tra -3 e +3. Questo numero rappresenta il numero di deviazioni standard che il parametro del giorno corrente rappresenta dalla media.

    S3 Storage Lens considera i parametri con un punteggio >2 o <-2 come valori anomali perché sono superiori o inferiori al 95% dei dati normalmente distribuiti.
  - Callout delle modifiche significative

    Il callout delle modifiche significative si applica ai parametri che dovrebbero cambiare meno frequentemente. Pertanto, è impostata su una sensibilità maggiore rispetto al calcolo delle anomalie, che in genere è compreso tra +/- 20% rispetto al giorno, alla settimana o al mese precedente.

    Risoluzione dei callout relativi ad archiviazione e utilizzo: se ricevi un callout di modifica significativa, non si tratta necessariamente di un problema. Tale call-out potrebbe essere il risultato di una modifica dello storage prevista. Ad esempio, è possibile che di recente sia stato

aggiunto un numero elevato di nuovi oggetti, eliminato un numero elevato di oggetti o apportate modifiche pianificate.

Se nel pannello di controllo viene visualizzato un callout di modifica significativa, prendine nota e determina se può essere spiegata dalle circostanze recenti. In caso contrario, utilizza il pannello di controllo di S3 Storage Lens per espandere ulteriori dettagli per comprendere le regioni, i bucket o i prefissi specifici che determinano la fluttuazione.

- Promemoria

  I promemoria forniscono informazioni dettagliate sul funzionamento di Amazon S3. Possono aiutarti a saperne di più sulle modalità di utilizzo delle funzionalità S3 per ridurre i costi di archiviazione o applicare best practice per la protezione dei dati.

Selezione dei parametri

S3 Storage Lens offre due selezioni di parametri che puoi scegliere per il tuo pannello di controllo e possono essere esportate: parametri gratuiti e raccomandazioni e parametri avanzati.

- Parametri gratuiti

  S3 Storage Lens offre parametri gratuiti per tutti i pannelli di controllo e le configurazioni. I parametri gratuiti contengono dati rilevanti per l'archiviazione, come il numero di bucket e gli oggetti nel tuo account. I parametri gratuiti includono anche parametri basati sui casi d'uso (ad esempio, parametri di ottimizzazione dei costi e protezione dei dati) che puoi utilizzare per verificare se l'archiviazione è configurata secondo le best practice di S3. Tutti i parametri gratuiti vengono raccolti quotidianamente. I dati sono disponibili per le query per 14 giorni. Per ulteriori informazioni sui parametri disponibili con i parametri gratuiti, consulta Glossario dei parametri di Amazon S3 Storage Lens.

- Raccomandazioni e parametri avanzati

  S3 Storage Lens offre parametri gratuiti per tutti i pannelli di controllo e le configurazioni con la possibilità di eseguire l'aggiornamento all'opzione di raccomandazioni e parametri avanzati. Vengono applicati costi aggiuntivi. Per ulteriori informazioni, consulta Prezzi di Amazon S3.

  I parametri avanzati e i suggerimenti includono tutti i parametri gratuiti e i parametri aggiuntivi, ad esempio i parametri avanzati relativi alla protezione dei dati e all'ottimizzazione dei costi, quelli relativi alle attività e quelli relativi ai codici di stato dettagliati. I parametri avanzati e i suggerimenti

forniscono anche suggerimenti per ottimizzare l'archiviazione. Le raccomandazioni vengono posizionate contestualmente insieme ai parametri pertinenti nel pannello di controllo.

Le raccomandazioni e i parametri avanzati includono le caratteristiche seguenti:

- Parametri avanzati: generano parametri aggiuntivi. Per un elenco completo delle categorie di parametri avanzati, consulta Categorie di parametri. Per un elenco completo di parametri, consulta Glossario dei parametri di Amazon S3 Storage Lens.
- Pubblicazione Amazon CloudWatch: consente di pubblicare i parametri di S3 Storage Lens relativi a utilizzo e attività in CloudWatch per creare una visualizzazione unificata dello stato operativo nei pannelli di controllo di CloudWatch. È inoltre possibile utilizzare le funzioni e le operazioni API di CloudWatch, come allarmi e azioni attivate, matematica dei parametri e rilevamento delle anomalie, per monitorare e intervenire sui parametri di S3 Storage Lens. Per ulteriori informazioni, consulta Monitoraggio dei parametri di S3 Storage Lens in CloudWatch.
- Aggregazione di prefisso: raccoglie i parametri a livello di prefisso. L'abilitazione dell'aggregazione di prefisso estende tutti i parametri inclusi nella configurazione del pannello di controllo a livello di prefisso. I parametri vengono generati solo per i prefissi che soddisfano la soglia configurata. Tieni presente che i parametri applicabili a livello di prefisso sono disponibili con l'Aggregazione di prefisso, ad eccezione delle impostazioni a livello di bucket e i parametri relativi al conteggio delle regole. I parametri a livello di prefisso non vengono pubblicati su CloudWatch.
- Aggregazione del gruppo Storage Lens: raccoglie i parametri a livello di gruppo Storage Lens. Dopo aver abilitato Raccomandazioni e i parametri avanzati e Aggregazione del gruppo di Storage Lens, puoi specificare quali gruppi di Storage Lens includere o escludere dal pannello di controllo di Storage Lens. Devi specificare almeno un gruppo Storage Lens. I gruppi Storage Lens specificati devono inoltre risiedere nella regione di origine designata nell'account del pannello di controllo. I parametri a livello di gruppo Storage Lens non vengono pubblicati su CloudWatch.

Tutti i parametri avanzati vengono raccolti quotidianamente. I dati sono disponibili per le query per al massimo 15 mesi. Per ulteriori informazioni sui parametri di archiviazione aggregati da S3 Storage Lens, consulta Glossario dei parametri di Amazon S3 Storage Lens.

> **Note**
>
> I suggerimenti sono disponibili solo quando si utilizza il pannello di controllo di S3 Storage Lens nella console di Amazon S3,

S3 Storage Lens e AWS Organizations

AWS Organizations è un Servizio AWS che consente di aggregare tutti gli Account AWS secondo una gerarchia organizzativa. Amazon S3 Storage Lens opera con AWS Organizations per fornire una vista unica dell'archiviazione e dell'attività di oggetti nello storage Amazon S3.

Per ulteriori informazioni, consulta Utilizzo di Amazon S3 Storage Lens con AWS Organizations.

- Accesso attendibile

  Utilizzando l'account di gestione dell'organizzazione, devi abilitare l'accesso attendibile per S3 Storage Lens per aggregare i parametri di archiviazione e i dati di utilizzo per tutti gli account membri dell'organizzazione. Puoi quindi creare pannelli di controllo o esportazioni per l'organizzazione utilizzando l'account di gestione o assegnando l'accesso da amministratore delegato ad altri account dell'organizzazione.

  Puoi disabilitare l'accesso attendibile per S3 Storage Lens in qualsiasi momento, evitando che S3 Storage Lens aggreghi i parametri per la tua organizzazione.
- Amministratore delegato

  Puoi creare pannelli di controllo e parametri per S3 Storage Lens per la tua organizzazione utilizzando l'account di gestione AWS Organizations o fornendo l'accesso come amministratore delegato ad altri account dell'organizzazione. Puoi annullare la registrazione degli amministratori delegati in qualsiasi momento. Questa azione interrompe automaticamente tutti i pannelli di controllo a livello di organizzazione creati dall'amministratore delegato dall'aggregazione di nuovi parametri di archiviazione.

Per maggiori informazioni, consulta Amazon S3 Storage Lens e AWS Organizations nella Guida per l'utente di AWS Organizations.

Ruoli collegati ai servizi per Amazon S3 Storage Lens

Oltre all'accesso attendibile di AWS Organizations, Amazon S3 Storage Lens utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo univoco di ruolo IAM collegato direttamente a S3 Storage Lens. I ruoli collegati ai servizi sono predefiniti da S3 Storage Lens e includono tutte le autorizzazioni necessarie per raccogliere i parametri di archiviazione e attività giornalieri dagli account membri dell'organizzazione.

Per ulteriori informazioni, consulta la sezione Utilizzo dei ruoli collegati ai servizi per Amazon S3 Storage Lens.

Utilizzo di Amazon S3 Storage Lens con AWS Organizations

Amazon S3 Storage Lens è una funzionalità di analisi dell'archiviazione su cloud che puoi utilizzare per avere una panoramica completa a livello di organizzazione sull'utilizzo e sulle attività relative all'archiviazione di oggetti. È possibile utilizzare i parametri di S3 Storage Lens per generare approfondimenti, ad esempio per scoprire la quantità di spazio di archiviazione disponibile nell'intera organizzazione o quali sono i bucket e i prefissi caratterizzati da una crescita più rapida. Puoi anche utilizzare i parametri di S3 Storage Lens per individuare le opportunità di ottimizzazione dei costi, implementare le best practice di protezione e sicurezza dei dati e migliorare le prestazioni dei carichi di lavoro delle applicazioni. Ad esempio, puoi identificare i bucket che non hanno regole del ciclo di vita S3 per far scadere i caricamenti in più parti incompleti che risalgono a più di 7 giorni. Puoi anche individuare i bucket non conformi alle best practice di protezione dei dati, come quelli che usano la replica S3 o il controllo delle versioni S3. S3 Storage Lens analizza i parametri di archiviazione per fornire raccomandazioni contestuali che puoi usare per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati.

Puoi utilizzare Amazon S3 Storage Lens per raccogliere i parametri di archiviazione e i dati di utilizzo per tutti gli Account AWS che fanno parte della tua gerarchia di AWS Organizations. A tale scopo, è necessario utilizzare AWS Organizations e abilitare l'accesso attendibile di S3 Storage Lens utilizzando l'account di gestione di AWS Organizations.

Dopo aver abilitato l'accesso attendibile, potrai aggiungere l'accesso da amministratore delegato agli account dell'organizzazione. Questi account possono quindi creare configurazioni e pannelli di controllo per S3 Storage Lens che raccolgono parametri di archiviazione a livello di organizzazione e dati utente.

Per maggiori informazioni sull'abilitazione dell'accesso attendibile, consulta Amazon S3 Storage Lens e AWS Organizations nella Guida per l'utente AWS Organizations.

Argomenti

Abilitazione dell'accesso attendibile per S3 Storage Lens

Abilitando l'accesso attendibile, consenti ad Amazon S3 Storage Lens di accedere alla tua gerarchia, appartenenza e struttura di AWS Organizations tramite le operazioni API AWS Organizations. S3 Storage Lens diventa in questo modo un servizio attendibile per l'intera struttura dell'organizzazione.

Ogni volta che viene creata una configurazione del pannello di controllo, S3 Storage Lens crea ruoli collegati ai servizi nella gestione o negli account dell'amministratore delegato. Il ruolo collegato ai servizi concede a S3 Storage Lens l'autorizzazione per eseguire le operazioni seguenti:

- Descrivere le organizzazioni
- Elencare gli account
- Verificare un elenco di accesso ai Servizio AWS per le organizzazioni
- Ottenere amministratori delegati per le organizzazioni

S3 Storage Lens può quindi garantire l'accesso per raccogliere i parametri tra account per gli account delle aziende. Per ulteriori informazioni, consulta la sezione Utilizzo dei ruoli collegati ai servizi per Amazon S3 Storage Lens.

Dopo aver abilitato l'accesso attendibile, potrai assegnare l'accesso da amministratore delegato agli account dell'organizzazione. Quando un account è contrassegnato come amministratore delegato per un servizio, l'account riceve l'autorizzazione ad accedere a tutte le operazioni API dell'organizzazione in sola lettura. Ciò fornisce visibilità di tipo amministratore delegato ai membri e alle strutture dell'organizzazione in modo che possano creare pannelli di controllo di S3 Storage Lens.

> **Note**
>
> Solo l'account di gestione può abilitare l'accesso attendibile per Amazon S3 Storage Lens.

Disabilitazione dell'accesso attendibile per S3 Storage Lens

Disabilitando l'accesso attendibile, si limita S3 Storage Lens al funzionamento solo a livello di account. Inoltre, ogni titolare dell'account può visualizzare solo le informazioni di S3 Storage Lens limitatamente all'ambito del proprio account e non all'intera organizzazione. Tutti i pannelli di controllo

che richiedono un accesso attendibile non saranno più aggiornati, ma conserveranno i dati della cronologia in base al periodo di disponibilità dei dati per le query.

> **Note**
>
> - Inoltre, la disabilitazione dell'accesso attendibile per S3 Storage Lens impedisce automaticamente a tutti i pannelli di controllo a livello di organizzazione di raccogliere e aggregare i parametri di storage.
> - Gli account amministratore e amministratore delegato saranno comunque in grado di visualizzare i dati della cronologia per i pannelli di controllo a livello organizzativo esistenti in base ai rispettivi periodi di conservazione per le query.

Registrazione di un amministratore delegato per S3 Storage Lens

È possibile creare pannelli di controllo a livello di organizzazione utilizzando l'account di gestione dell'organizzazione o un account come amministratore delegato. Gli account amministratore delegati consentono ad altri account oltre all'account di gestione di creare pannelli di controllo a livello di organizzazione. Solo l'account di gestione di un'organizzazione può registrare e annullare la registrazione di altri account come amministratori delegati per l'organizzazione.

Per registrare un amministratore delegato utilizzando la console di Amazon S3, consulta Registrazione di amministratori delegati per S3 Storage Lens.

Puoi inoltre registrare un amministratore delegato utilizzando la REST API AWS Organizations, la AWS CLI o gli SDK dall'account di gestione. Per ulteriori informazioni, consulta RegisterDelegatedAdministrator nella documentazione di riferimento dell'API AWS Organizations.

> **Note**
>
> Prima di poter designare un amministratore delegato utilizzando la REST API AWS Organizations, la AWS CLI o gli SDK, è necessario chiamare l'operazione EnableAWSOrganizationsAccess.

Annullamento della registrazione di un amministratore delegato per S3 Storage Lens

Puoi annullare la registrazione di un account amministratore delegato. Gli account amministratore delegati consentono ad altri account oltre all'account di gestione di creare pannelli di controllo a livello di organizzazione. Solo l'account di gestione di un'organizzazione può annullare la registrazione degli account come amministratori delegati per l'organizzazione.

Per annullare la registrazione di un amministratore delegato tramite la console S3, consulta Annullamento della registrazione di amministratori delegati per S3 Storage Lens.

Puoi inoltre annullare la registrazione di un amministratore delegato utilizzando la REST API AWS Organizations, la AWS CLI o gli SDK dall'account di gestione. Per ulteriori informazioni, consulta DeregisterDelegatedAdministrator nella documentazione di riferimento dell'API AWS Organizations.

> **Note**
>
> - Questa azione interrompe automaticamente tutti i pannelli di controllo a livello di organizzazione creati dall'amministratore delegato dall'aggregazione di nuovi parametri di archiviazione.
> - Gli account amministratore delegato con registrazione annullata saranno comunque in grado di visualizzare i dati della cronologica di tali pannelli di controllo mentre i dati sono disponibili per le query.

Autorizzazioni Amazon S3 Storage Lens

Amazon S3 Storage Lens richiede nuove autorizzazioni in AWS Identity and Access Management(IAM) per autorizzare l'accesso alle operazioni S3 Storage Lens. Per concedere queste autorizzazioni, puoi utilizzare una policy IAM basata sull'identità. Per farlo, collega la policy a utenti, gruppi o ruoli IAM ai quali desideri concedere tali autorizzazioni. Le autorizzazioni possono riguardare, ad esempio, l'abilitazione o la disabilitazione di S3 Storage Lens, l'accesso a qualsiasi pannello di controllo o la configurazione di S3 Storage Lens.

L'utente o il ruolo IAM deve appartenere all'account del creatore o del titolare del pannello di controllo o della configurazione, a meno che non si verifichino entrambe le condizioni seguenti:

- Il tuo account è membro di AWS Organizations.

- Ti è stata concessa l'autorizzazione per creare pannelli di controllo a livello di organizzazione dal tuo account di gestione in qualità di amministratore delegato.

> **Note**
>
> - Non puoi utilizzare le credenziali utente root del tuo account per visualizzare i pannelli di controllo di Amazon S3 Storage Lens. Per accedere ai pannelli di controllo di S3 Storage Lens, è necessario concedere le autorizzazioni IAM necessarie a un utente IAM nuovo o esistente. Quindi, accedi con le credenziali utente per accedere ai pannelli di controllo di S3 Storage Lens. Per ulteriori informazioni, consulta Best Practice di sicurezza in IAM nella Guida per l'utente di IAM.
> - L'utilizzo di S3 Storage Lens nella console di Amazon S3 può richiedere più autorizzazioni. Ad esempio, per modificare un pannello di controllo nella console, sono necessarie le seguenti autorizzazioni:
>   - `s3:ListStorageLensConfigurations`
>   - `s3:GetStorageLensConfiguration`
>   - `s3:PutStorageLensConfiguration`

Argomenti

Impostazione delle autorizzazioni dell'account per utilizzare S3 Storage Lens

Per creare e gestire i pannelli di controllo di S3 Storage Lens e le configurazioni dei pannelli di controllo di Storage Lens, devi disporre delle seguenti autorizzazioni, a seconda delle azioni che desideri eseguire:

Autorizzazioni IAM relative a Amazon S3 Storage Lens

Operazione	Autorizzazioni IAM
Creare o aggiornare un pannello di controllo di S3 Storage Lens nella console di Amazon S3.	`s3:ListStorageLensConfigurations` `s3:GetStorageLensConfiguration` `s3:GetStorageLensConfigurationTagging` `s3:PutStorageLensConfiguration` `s3:PutStorageLensConfigurationTagging`
Ottenere i tag di un pannello di controllo di S3 Storage Lens nella console di Amazon S3.	`s3:ListStorageLensConfigurations` `s3:GetStorageLensConfigurationTagging`
Visualizzare un pannello di controllo di S3 Storage Lens nella console di Amazon S3.	`s3:ListStorageLensConfigurations` `s3:GetStorageLensConfiguration` `s3:GetStorageLensDashboard`
Eliminare un pannello di controllo di S3 Storage Lens nella console di Amazon S3.	`s3:ListStorageLensConfigurations` `s3:GetStorageLensConfiguration` `s3:DeleteStorageLensConfiguration`
Creare o aggiornare una configurazione S3 Storage Lens mediante la AWS CLI o un SDK AWS.	`s3:PutStorageLensConfiguration` `s3:PutStorageLensConfigurationTagging`
Ottenere i tag di una configurazione S3 Storage Lens mediante la AWS CLI o un SDK AWS.	`s3:GetStorageLensConfigurationTagging`

Operazione	Autorizzazioni IAM
Visualizzare una configurazione S3 Storage Lens mediante la AWS CLI o un SDK AWS.	`s3:GetStorageLensConfiguration`
Eliminare una configurazione S3 Storage Lens mediante la AWS CLI o un SDK AWS.	`s3:DeleteStorageLensConfiguration`

Note

- Puoi utilizzare i tag delle risorse in una policy IAM per gestire le autorizzazioni.
- Un ruolo o un utente IAM con queste autorizzazioni può visualizzare i parametri dai bucket e dai prefissi in cui potrebbero non disporre dell'autorizzazione diretta per leggere o elencare oggetti.
- Per i pannelli di controllo di S3 Storage Lens con parametri a livello di prefisso abilitati, se un percorso di prefisso selezionato corrisponde a una chiave di oggetto, il pannello di controllo potrebbe visualizzare la chiave di oggetto come un altro prefisso.
- Per le esportazioni dei parametri, archiviati in un bucket dell'account, le autorizzazioni vengono concesse mediante l'autorizzazione `s3:GetObject` esistente nella policy IAM. Analogamente, per un'entità AWS Organizations, l'account di gestione dell'organizzazione o l'amministratore delegato può utilizzare le policy IAM per gestire le autorizzazioni di accesso per pannelli di controllo e configurazioni a livello di organizzazione.

Impostazione delle autorizzazioni dell'account per utilizzare i gruppi S3 Storage Lens

Puoi utilizzare i gruppi S3 Storage Lens per comprendere la distribuzione dell'archiviazione all'interno dei bucket in base al prefisso, al suffisso, al tag dell'oggetto, alla dimensione dell'oggetto o all'età dell'oggetto. Per visualizzare i parametri aggregati, collega i gruppi Storage Lens al pannello di controllo.

Per utilizzare i gruppi Storage Lens, sono necessarie autorizzazioni specifiche. Per ulteriori informazioni, consulta the section called “Autorizzazioni gruppi Storage Lens”.

Impostazione delle autorizzazioni per utilizzare S3 Storage Lens con AWS Organizations

È possibile utilizzare Amazon S3 Storage Lens per raccogliere i parametri di archiviazione e i dati di utilizzo per tutti gli account che fanno parte della gerarchia di AWS Organizations. Di seguito sono riportate le operazioni e le autorizzazioni relative all'utilizzo di S3 Storage Lens con Organizations.

AWS OrganizationsAutorizzazioni IAM relative a per l'utilizzo di S3 Storage Lens

Operazione	Autorizzazioni IAM
Abilitare l'accesso attendibile per S3 Storage Lens per la tua organizzazione.	`organizations:EnableAWSServiceAccess`
Disabilitare l'accesso attendibile per S3 Storage Lens per la tua organizzazione.	`organizations:DisableAWSServiceAccess`
Registrare un amministratore delegato per creare pannelli di controllo o configurazioni S3 Storage Lens per l'organizzazione.	`organizations:RegisterDelegatedAdministrator`
Annullare la registrazione di un amministratore delegato in modo che non possa più creare pannelli di controllo o configurazioni di S3 Storage Lens per l'organizzazione.	`organizations:DeregisterDelegatedAdministrator`
Autorizzazioni aggiuntive per creare configurazioni S3 Storage Lens a livello di organizzazione.	`organizations:DescribeOrganization` `organizations:ListAccounts` `organizations:ListAWSServiceAccessForOrganization` `organizations:ListDelegatedAdministrators` `iam:CreateServiceLinkedRole`

Visualizzazione dei parametri con Amazon S3 Storage Lens

S3 Storage Lens aggrega i tuoi parametri e mostra le informazioni nella sezione Account snapshot (Snapshot dell'account) nella pagina Buckets (Bucket) della console Amazon S3. S3 Storage Lens dispone inoltre di un pannello di controllo interattivo che puoi usare per visualizzare informazioni dettagliate e tendenze, contrassegnare le anomalie e ricevere consigli per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati. Nel pannello di controllo sono disponibili opzioni di drill-down per generare e visualizzare approfondimenti a livello di organizzazione, account, Regione AWS, classe di storage, bucket, prefisso o gruppo Storage Lens. Puoi anche inviare un'esportazione dei parametri quotidiana in formato CSV o Parquet a un bucket S3.

Per impostazione predefinita, tutti pannelli di controllo sono configurati con parametri gratuiti, che includono i parametri che puoi utilizzare per comprendere l'utilizzo e l'attività dell'archiviazione S3, ottimizzare i costi di archiviazione e implementare le best practice per la protezione dei dati e la gestione degli accessi. I parametri gratuiti vengono aggregate fino al livello del bucket. Con i parametri gratuiti, i dati sono disponibili per le query per un massimo di 14 giorni.

I parametri avanzati e i suggerimenti includono le seguenti funzionalità aggiuntive che puoi utilizzare per ottenere ulteriori informazioni sull'utilizzo e sulle attività a livello di archiviazione, nonché le best practice per ottimizzarlo:

- Suggerimenti contestuali (disponibili solo nel pannello di controllo)
- Parametri avanzati (inclusi i parametri delle attività aggregati per bucket)
- Aggregazione di prefisso
- Aggregazione di un gruppo Storage Lens
- Aggregazione di un gruppo Storage Lens
- Pubblicazione Amazon CloudWatch

I dati dei parametri avanzati sono disponibili per le query per 15 mesi. Per l’uso di S3 Storage Lens con i parametri avanzati sono previsti costi aggiuntivi. Per ulteriori informazioni, consulta Prezzi di Amazon S3. Per ulteriori informazioni su parametri gratuiti e avanzati, consulta Selezione dei parametri.

Argomenti

- Visualizzazione dei parametri di S3 Storage Lens nei pannelli di controllo

- Visualizzazione dei parametri di Amazon S3 Storage Lens utilizzando una esportazione di dati
- Monitoraggio dei parametri di S3 Storage Lens in CloudWatch

Visualizzazione dei parametri di S3 Storage Lens nei pannelli di controllo

Nella console di Amazon S3, Storage Lens S3 fornisce un pannello di controllo interattivo predefinito con il quale è possibile visualizzare approfondimenti e tendenze relative ai dati. È possibile utilizzare il pannello di controllo per visualizzare informazioni dettagliate e tendenze, contrassegnare le anomalie e ricevere suggerimenti per ottimizzare i costi di archiviazione e applicare best practice per la protezione dei dati. Nel pannello di controllo sono disponibili opzioni di drill-down per generare e visualizzare approfondimenti a livello di account, bucket, Regione AWS, prefisso o gruppo Storage Lens. Se S3 Storage Lens è abilitato per l'utilizzo con AWS Organizations, è anche possibile generare approfondimenti a livello di organizzazione (come i dati per tutti gli account che fanno parte della gerarchia AWS Organizations). Il pannello di controllo viene sempre caricato per la data più recente per la quale sono disponibili i parametri.

Il pannello di controllo predefinito di S3 Storage Lens nella console è denominato default-account-dashboard. Amazon S3 preconfigura questo pannello di controllo per visualizzare gli approfondimenti di riepilogo e le tendenze per l'intero account e le aggiorna quotidianamente nella console S3. Non è possibile modificare l'ambito di configurazione del pannello di controllo predefinito, ma è possibile aggiornare la selezione dei parametri, dai parametri gratuiti alle raccomandazioni e ai parametri avanzati a pagamento. I parametri avanzati e i suggerimenti ti consentono di accedere a parametri e funzionalità aggiuntive. Queste funzionalità includono categorie di parametri avanzati, aggregazione a livello di prefisso, suggerimenti contestuali e funzionalità di pubblicazione Amazon CloudWatch.

Puoi disabilitare il pannello di controllo predefinito, ma non puoi eliminarlo. In caso di disattivazione del pannello di controllo predefinito, non viene più aggiornato. Non riceverai più alcun nuovo parametro giornaliero in S3 Storage Lens o nella sezione Snapshot dell'account nella pagina Buckets. Puoi comunque visualizzare i dati della cronologia nel pannello di controllo predefinito fino alla scadenza delle query di dati (14 giorni). Questo periodo è di 15 mesi se hai abilitato i parametri avanzati e i suggerimenti. Per accedere a questi dati, puoi riabilitare il pannello di controllo predefinito entro il periodo di scadenza.

Puoi inoltre creare altri pannelli di controllo di S3 Storage Lens e definirne l'ambito per Regioni AWS, bucket S3 o account. Se hai abilitato l'utilizzo di Storage Lens con AWS Organizations, puoi anche definire l'ambito dei pannelli di controllo per organizzazione. Quando crei o modifichi un pannello di controllo di S3 Storage Lens, ne definisci l'ambito e la selezione dei parametri.

Puoi disabilitare o eliminare eventuali pannelli di controllo aggiuntivi creati.

- Se disattivi un pannello di controllo, questo non sarà più aggiornato e non riceverai più nuovi parametri giornalieri. Puoi comunque visualizzare i dati della cronologia per i parametri gratuiti fino al periodo di scadenza di 14 giorni. Se hai abilitato i parametri avanzati e i suggerimenti per il pannello di controllo, questo periodo è di 15 mesi. Per accedere a questi dati, puoi riabilitare il pannello di controllo entro il periodo di scadenza.
- Se si elimina il pannello di controllo, tutte le impostazioni di configurazione del pannello di controllo saranno perse. Non riceverai più nuovi parametri giornalieri e perderai anche l'accesso ai dati della cronologia associati a tale pannello di controllo. Se desideri accedere ai dati della cronologia di un pannello di controllo eliminato, dovrai creare un altro pannello di controllo con lo stesso nome nella stessa regione di origine.

Argomenti

Visualizzazione di un pannello di controllo di Amazon S3 Storage Lens

La seguente procedura descrive come visualizzare un pannello di controllo di S3 Storage Lens nella console S3. Per le procedure dettagliate basate sui casi d'uso che mostrano come utilizzare il pannello di controllo per ottimizzare i costi di archiviazione, implementare le best practice e migliorare le prestazioni delle applicazioni che accedono ai bucket S3, consulta Casi d'uso relativi ai parametri di Amazon S3 Storage Lens.

> **Note**
>
> Non puoi utilizzare le credenziali utente root del tuo account per visualizzare i pannelli di controllo di Amazon S3 Storage Lens. Per accedere ai pannelli di controllo di S3 Storage Lens, è necessario concedere le autorizzazioni AWS Identity and Access Management (IAM) necessarie a un utente IAM nuovo o esistente. Quindi, accedi con le credenziali utente per accedere ai pannelli di controllo di S3 Storage Lens. Per ulteriori informazioni, consulta Autorizzazioni Amazon S3 Storage Lens e Best Practice di sicurezza in IAM nella Guida per l'utente di IAM.

1. Accedi alla AWS Management Console e apri la console di Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Pannelli di controllo, scegli il pannello di controllo che desideri visualizzare.

   Il pannello di controllo si apre in S3 Storage Lens. La sezione Snapshot for date (Snapshot per [data]) mostra l'ultima data in cui S3 Storage Lens ha raccolto i parametri. Il pannello di controllo viene sempre caricato alla data più recente per la quale sono disponibili i parametri.
4. (Facoltativo) Per modificare la data del pannello di controllo di S3 Storage Lens, nel selettore della data in alto a destra, scegli una nuova data.
5. (Facoltativo) Per applicare filtri temporanei per limitare ulteriormente l'ambito dei dati del pannello di controllo, procedi come segue:

   a. Espandi la sezione Filtri.

   b. Per filtrare per account, Regioni AWS, classi di archiviazione, bucket, prefissi o gruppi Storage Lens specifici scegli le opzioni in base a cui filtrare.

   > **Note**
   >
   > Il filtro Prefissi e il filtro Gruppi Storage Lens non possono essere applicati contemporaneamente.

   c. Per aggiornare un filtro, scegli Apply (Applica).

   d. Per rimuovere un filtro, fai clic sulla X accanto al filtro.
6. In qualsiasi sezione del pannello di controllo di S3 Storage Lens, per visualizzare i dati relativi a un parametro specifico, in Metric (Parametro), scegli il nome del parametro.
7. In qualsiasi grafico o visualizzazione del pannello di controllo di S3 Storage Lens, puoi eseguire il drill-down dei livelli di aggregazione utilizzando le schede Account, Regioni AWS, Classi di archiviazione, Bucket, Prefissi, o Gruppi Storage Lens Per vedere un esempio, consulta Scoperta dei bucket Amazon S3 freddi.

Informazioni sul pannello di controllo di S3 Storage Lens

Il pannello di controllo di S3 Storage Lens è costituito da una scheda Overview (Panoramica) principale e da un massimo di cinque schede aggiuntive che rappresentano ogni livello di aggregazione:

- Account
- Regioni AWS
- Classi di archiviazione
- Bucket
- Prefissi
- Gruppi Storage Lens

Nella scheda Overview (Panoramica), i dati del pannello di controllo vengono aggregati in tre diverse sezioni: Snapshot for date (Snapshot per [data]), Trends and distributions (Tendenze e distribuzioni) e Top N overview (Panoramica primi N).

Per ulteriori informazioni sul pannello di controllo di S3 Storage Lens, consulta le sezioni seguenti.

Snapshot

La sezione Snapshot for date (Snapshot per [data]) mostra i parametri di riepilogo aggregati da S3 Storage Lens per la data selezionata. Questi parametri di riepilogo includono i seguenti parametri:

- Archiviazione totale: la quantità di archiviazione totale utilizzata in byte.
- Conteggio oggetti: il numero totale di oggetti nel tuo Account AWS.
- Dimensione media oggetto: dimensione media dell'oggetto.
- Bucket attivi: numero totale di bucket attivi in uso nel tuo account con archiviazione >0 byte.
- Account: numero di account la cui archiviazione è compresa nell'ambito. Questo valore è 1 a meno che non si utilizzi AWS Organizations e S3 Storage Lens abbia un accesso attendibile con un ruolo collegato ai servizi valido. Per ulteriori informazioni, consulta Utilizzo dei ruoli collegati ai servizi per Amazon S3 Storage Lens.
- Bucket: il numero totale di bucket nel tuo account.

Dati del parametro

Per ogni parametro visualizzato nello snapshot, puoi visualizzare i seguenti dati:

- Nome parametro: nome del parametro.
- Categoria parametro: categoria in cui è incluso il parametro.
- Totale per data: conteggio totale per la data selezionata.

- % variazione: variazione percentuale rispetto alla data dell'ultimo snapshot.
- Tendenza a 30 giorni: linea di tendenza che mostra le variazioni del parametro in un periodo di 30 giorni.
- Suggerimento: suggerimento contestuale basato sui dati forniti nello snapshot. I suggerimenti sono disponibili con i parametri avanzati e i suggerimenti. Per ulteriori informazioni, consulta Raccomandazioni.

Categorie di parametri

Facoltativamente, puoi aggiornare la sezione Snapshot for date (Snapshot per [data]) del pannello di controllo per visualizzare i parametri di altre categorie. Se desideri visualizzare i dati snapshot per altri parametri, puoi scegliere altri valori in Metrics categories (Categorie parametri):

- Ottimizzazione dei costi
- Protezione dei dati
- Attività (disponibile con i parametri avanzati)
- Gestione degli accessi
- Prestazioni
- Eventi

La sezione Snapshot for date (Snapshot per [data]) mostra solo una selezione di parametri per ogni categoria. Per visualizzare tutti i parametri di una categoria specifica, scegli il parametro desiderato nelle sezioni Trends and distributions (Tendenze e distribuzioni) o Top N overview (Panoramica primi N). Per ulteriori informazioni sulle categorie di parametri, consulta Categorie di parametri. Per un elenco completo dei parametri di S3 Storage Lens, consulta Glossario dei parametri di Amazon S3 Storage Lens.

Trends and distributions (Tendenze e distribuzione)

La seconda sezione della scheda Overview (Panoramica) è Trends and distributions (Tendenze e distribuzione). Nella sezione Trends and distributions (Tendenze e distribuzioni), puoi scegliere due parametri da confrontare in un intervallo di date definito. La sezione Trends and distributions (Tendenze e distribuzioni) mostra la relazione tra due parametri nel tempo. In questa sezione sono vsualizzati i grafici che puoi utilizzare per visualizzare le distribuzioni Storage class (Classe di archiviazione) e Region (Regione) tra le due tendenze che stai monitorando. Facoltativamente, puoi eseguire il drill-down su un punto dati in uno dei grafici per un'analisi più approfondita.

Per una procedura dettagliata che utilizza la sezione Trends and distributions (Tendenze e distribuzioni), consulta Identificare i bucket che non utilizzano la crittografia lato server con AWS KMS per la crittografia predefinita (SSE-KMS).

Panoramica Top N

La terza sezione del pannello di controllo di S3 Storage Lens è Panoramica Top N (ordinata in ordine crescente o decrescente). Questa sezione mostra i parametri selezionati tra i primi N account, Regioni AWS, bucket, prefissi o gruppi Storage Lens. Se S3 Storage Lens è abilitato per l'utilizzo con AWS Organizations, puoi anche vedere i parametri selezionati nell'organizzazione.

Per una procedura dettagliata che utilizza la sezione Top N overview (Panoramica primi N), consulta Identificare i bucket S3 più grandi.

Drill-down e analisi per opzioni

Per fornire un'esperienza fluida nell'esecuzione delle analisi, il pannello di controllo di S3 Storage Lens offre un menu delle azioni, che viene visualizzato quando si sceglie un valore del grafico. Per utilizzare questo menu, seleziona un qualsiasi valore del grafico e scegli tra le due opzioni disponibili nella casella:

- L'azione Drill down (Drill-down) applica il valore selezionato come filtro in tutte le schede del pannello di controllo. Puoi quindi eseguire il drill-down in tale valore per un'analisi più approfondita.
- L'azione Analizza per consente di accedere alla scheda Dimensione selezionata e di applicare tale valore come filtro. Queste schede includono Account, Regioni AWS, Classi di archiviazione, Bucket, Prefissi (per pannelli di controllo con Parametri avanzati e Aggregazione prefissi abilitati) e Gruppi Storage Lens (per i pannelli di controllo con Parametri avanzati e Aggregazione gruppi Storage Lens abilitati). Analizza per ti permette di vedere i dati nel contesto della nuova dimensione per un'analisi più approfondita.

Le azioni Drill down e Analizza per potrebbero essere disabilitate se il risultato dovesse produrre valori illogici o nessun valore. Entrambe le azioni Drill down e Analizza per determinano l'applicazione di filtri oltre a qualsiasi altro filtro esistente in tutte le schede del pannello di controllo. È inoltre possibile rimuovere filtri in base alle esigenze.

Schede

Le schede a livello di dimensione forniscono una vista dettagliata di tutti i valori all'interno di una determinata dimensione. Ad esempio, la scheda Regioni AWS mostra i parametri per tutte le Regioni

AWS, mentre la scheda Bucket mostra i parametri per tutti i bucket. Ogni scheda dimensione contiene un layout identico costituito da quattro sezioni:

- Un grafico delle tendenze che mostra i primi N elementi all'interno della dimensione negli ultimi 30 giorni per il parametro selezionato. Per impostazione predefinita, questo grafico visualizza i primi 10 elementi, ma è possibile ridurli ad almeno 3 elementi o aumentarli fino a 50 elementi.
- Un istogramma mostra un grafico a barre verticali per la data e il parametro selezionati. Se è presente un numero molto elevato di elementi da visualizzare in questo grafico, potrebbe essere necessario scorrerlo orizzontalmente.
- Un diagramma di analisi a bolle che riporta tutti gli elementi che rientrano in quella dimensione. Questo grafico rappresenta il primo parametro sull'asse x e il secondo parametro sull'asse y. Il terzo parametro è rappresentato dalla dimensione della bolla.
- Una visualizzazione a griglia dei parametri che contiene ogni elemento della dimensione elencata nelle righe. Le colonne rappresentano ogni parametro disponibile, disposti in schede delle categorie di parametri per facilitare la navigazione.

Visualizzazione dei parametri di Amazon S3 Storage Lens utilizzando una esportazione di dati

I parametri di Amazon S3 Storage Lens vengono generati quotidianamente in file di esportazione di parametri in formato CSV o Apache Parquet e inseriti in un bucket S3 nel tuo account. Da lì, puoi inserire le metriche esportate negli strumenti di analisi di tua scelta, come Amazon QuickSight e Amazon Athena, dove puoi analizzare l'utilizzo dello storage e le tendenze delle attività.

Argomenti

Utilizzo di un file AWS KMS key per crittografare le esportazioni delle metriche

Per concedere ad Amazon S3 Storage Lens l'autorizzazione alla crittografia delle esportazioni dei parametri mediante una chiave gestita dal cliente, devi utilizzare una policy di chiave. Per aggiornare la policy di chiave in modo da poter utilizzare una chiave KMS per crittografare le esportazioni dei parametri di S3 Storage Lens, segui la seguente procedura.

Per concedere le autorizzazioni di S3 Storage Lens per eseguire la crittografia dei dati utilizzando la chiave KMS

1. Accedi a AWS Management Console utilizzando la chiave gestita dal cliente Account AWS che possiede.
2. Apri la AWS KMS console all'indirizzo https://console.aws.amazon.com/kms.
3. Per modificare la Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
4. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
5. In Chiavi gestite dal cliente, scegli la chiave che desideri utilizzare per crittografare le esportazioni delle metriche. AWS KMS keys sono specifici della regione e devono trovarsi nella stessa regione del bucket S3 di destinazione di esportazione delle metriche.
6. In Policy chiave, seleziona Passa alla visualizzazione della policy.
7. Per aggiornare la policy chiave, seleziona Modifica.
8. In Modifica policy della chiave, aggiungi la policy chiave seguente alla policy chiave esistente. Per utilizzare questa policy, sostituisci *`user input placeholders`* con le tue informazioni.

```
{
    "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. Seleziona Salvataggio delle modifiche.

Per ulteriori informazioni sulla creazione di chiavi gestite dal cliente e sull'utilizzo delle policy delle chiavi, consulta i seguenti argomenti nella Guida per Developer di AWS Key Management Service :

- Nozioni di base
- Utilizzo delle politiche chiave in AWS KMS

Puoi anche utilizzare la AWS KMS PUT key policy API operation (PutKeyPolicy) per copiare la policy chiave nelle chiavi gestite dai clienti che desideri utilizzare per crittografare le esportazioni delle metriche utilizzando l'API REST e gli AWS CLI SDK.

Cos'è un manifest di esportazione di S3 Storage Lens?

Data la grande quantità di dati aggregati, una esportazione giornaliera di parametri di S3 Storage Lens può essere suddivisa in più file. Il file manifest `manifest.json` descrive dove si trovano i file di esportazione dei parametri quel giorno. Ogni volta che viene consegnata una nuova esportazione, questa è accompagnata da un nuovo manifest. Ogni manifest contenuto nel file `manifest.json` fornisce i metadati e altre informazioni di base riguardanti un inventario.

Le informazioni sul manifest includono le seguenti proprietà:

- `sourceAccountId`: l'ID account del proprietario della configurazione.
- `configId`: un identificativo univoco per il pannello di controllo.
- `destinationBucket`: l'ARN (Amazon Resource Name) del bucket di destinazione in cui viene inserita l'esportazione dei parametri.
- `reportVersion`: la versione dell'esportazione.
- `reportDate`: la data del report.
- `reportFormat`: il formato del report.
- `reportSchema`: lo schema del report.
- `reportFiles`: l'elenco reale dei file di report di esportazione presenti nel bucket di destinazione.

Di seguito viene riportato un esempio di un manifest in un file `manifest.json` per una esportazione in 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"
        }
      ]
}
```

Di seguito viene riportato un esempio di un manifesto in un file `manifest.json` per una esportazione in 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"
      }
}
```

Puoi configurare l'esportazione delle metriche in modo che venga generata come parte della configurazione del dashboard nella console Amazon S3 o utilizzando l'API REST AWS CLI e gli SDK di Amazon S3.

Informazioni sullo schema di esportazione di Amazon S3 Storage Lens

La tabella seguente contiene lo schema di esportazione dei parametri di S3 Storage Lens.

Nome attributo	Tipo di dati	Nome colonna	Descrizione
`VersionNumber`	Stringa	`version_number`	La versione dei parametri di S3 Storage Lens in uso.
`ConfigurationId`	Stringa	`configuration_id`	`configuration_id` della configurazione di S3 Storage Lens.
`ReportDate`	Stringa	`report_date`	Data in cui sono stati tracciati i parametri.
`AwsAccountNumber`	Stringa	`aws_account_number`	Il tuo numero. Account AWS
`AwsRegion`	Stringa	`aws_region`	La metrica Regione AWS per cui vengono tracciate le metriche.
`StorageClass`	Stringa	`storage_class`	La classe di storage del bucket in questione.
`RecordType`	ENUM	`record_type`	Il tipo di artefatto che viene riportato (ACCOUNT, BUCKET o PREFISSO).
`RecordValue`	Stringa	`record_value`	Il valore dell'artefatto `RecordType` .

Nome attributo	Tipo di dati	Nome colonna	Descrizione
			Note Il `record_value` è codificato in formato URL.
`BucketName`	Stringa	`bucket_name`	Il nome del bucket che viene riportato.
`MetricName`	Stringa	`metric_name`	Il nome del parametro che viene riportato.
`MetricValue`	Lungo	`metric_value`	Il valore del parametro che viene riportato.

Esempio di esportazione dei parametri di S3 Storage Lens

Di seguito è riportato un esempio di esportazione dei parametri di S3 Storage Lens basata su questo schema.

Note

Per identificare i parametri per i gruppi Storage Lens, cerca il valore `STORAGE_LENS_GROUP_BUCKET` o `STORAGE_LENS_GROUP_ACCOUNT` nella colonna `record_type`. La colonna `record_value` mostra il nome della risorsa Amazon (ARN) per il gruppo Storage Lens, ad esempio, `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

Di seguito è riportato un esempio di esportazione dei parametri di S3 Storage Lens con i dati dei gruppi di Storage Lens.

version_numbe	configuration_id	report_date	aws_account_num	aws_region	storage_class	record_type	record_value	bucket_name	metric_name	metric_value
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		StorageBytes	3128548856
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ObjectCount	4440
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		EncryptedStorageBytes	3128548856
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		EncryptedObjectCount	4440
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		DeleteMarkerObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ObjectLockEnabledStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ObjectLockEnabledObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		CurrentVersionStorageBytes	3128548856
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		CurrentVersionObjectCount	4440
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		NonCurrentVersionStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		NonCurrentVersionObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMultipartUploadStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMultipartUploadObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		DeleteMarkerStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedStorageBytesSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedObjectCountSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMPUStorageBytesOlderThan7Days	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMPUObjectCountOlderThan7Days	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	StorageBytes	676863200
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ObjectCount	3000
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	EncryptedStorageBytes	676863200
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	EncryptedObjectCount	3000
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	DeleteMarkerObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ObjectLockEnabledStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ObjectLockEnabledObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	CurrentVersionStorageBytes	676863200
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	CurrentVersionObjectCount	3000
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	NonCurrentVersionStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	NonCurrentVersionObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMultipartUploadStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMultipartUploadObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	DeleteMarkerStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedStorageBytesSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedObjectCountSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMPUStorageBytesOlderThan7Days	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMPUObjectCountOlderThan7Days	0

Monitoraggio dei parametri di S3 Storage Lens in CloudWatch

Puoi pubblicare i parametri di S3 Storage Lens su Amazon CloudWatch per creare una visualizzazione unificata dello stato operativo nei pannelli di controllo di CloudWatch. È inoltre possibile utilizzare le funzioni di CloudWatch, come allarmi e azioni attivate, matematica dei parametri e rilevamento delle anomalie, per monitorare e intervenire sui parametri di S3 Storage Lens. Inoltre, le operazioni API di CloudWatch consentono alle applicazioni, inclusi i provider di terze

parti, di accedere ai parametri di S3 Storage Lens. Per ulteriori informazioni sulle caratteristiche di CloudWatch, consulta la Guida per l'utente di Amazon CloudWatch.

Puoi abilitare l'opzione di pubblicazione CloudWatch per pannelli di controllo nuovi o già esistenti utilizzando la console Amazon S3, la REST API Amazon S3, la AWS CLI e gli SDK AWS. I pannelli di controllo che sono stati aggiornati su Raccomandazioni e parametri avanzati di S3 Storage Lens possono utilizzare l'opzione di pubblicazione di CloudWatch. Per i prezzi relativi all'opzione Advanced metrics and recommendations (Parametri avanzati e suggerimenti), consulta Prezzi di Amazon S3. Non vengono applicati ulteriori addebiti di pubblicazione dei parametri CloudWatch; tuttavia, sono applicabili altri costi di CloudWatch come pannelli di controllo, allarmi e chiamate API. Per ulteriori informazioni, consulta Prezzi di Amazon CloudWatch.

I parametri S3 Storage Lens sono pubblicati su CloudWatch nell'account che possiede la configurazione di S3 Storage Lens. Dopo aver abilitato l'opzione di pubblicazione CloudWatch all'interno dei parametri avanzati e suggerimenti, puoi accedere ai parametri a livello di organizzazione, account e bucket in CloudWatch. I parametri a livello di prefissi non sono disponibili su CloudWatch.

> **Note**
>
> I parametri S3 Storage Lens sono parametri giornalieri e vengono pubblicati su CloudWatch una volta al giorno. Quando si interrogano i parametri S3 Storage Lens in CloudWatch, il periodo per la query deve essere di 1 giorno (86.400 secondi). Dopo che i parametri quotidiani di S3 Storage Lens sono visualizzati nel pannello di controllo di S3 Storage Lens nella console Amazon S3, possono essere necessarie alcune ore perché questi parametri siano visualizzati su CloudWatch. Quando abiliti per la prima volta l'opzione di pubblicazione di CloudWatch per i parametri S3 Storage Lens, possono essere necessarie fino a 24 ore prima che i tuoi parametri vengano pubblicati su CloudWatch.

Dopo aver abilitato l'opzione di pubblicazione di CloudWatch, è possibile utilizzare le seguenti funzionalità CloudWatch per monitorare e analizzare i dati di S3 Storage LensStorage Lens:

- Pannelli di controllo – Utilizza i pannelli di controllo CloudWatch per creare pannelli di controllo personalizzati di S3 Storage Lens. Condividi il tuo pannello di controllo di CloudWatch con persone che non hanno accesso diretto al tuo Account AWS, tra i team, con le parti interessate e con persone esterne alle tue organizzazioni.

- Allarmi e operazioni attivate – Configura gli allarmi che controllano i parametri e si attivano quando viene violata una soglia. Ad esempio, è possibile configurare un allarme che invia una notifica di Amazon SNS quando il valore del parametro Incomplete Multipart Upload Bytes (Byte caricamenti in più parti incompleti) supera 1 GB per tre giorni consecutivi.
- Rilevamento di anomalie – Abilita il rilevamento delle anomalie per analizzare continuamente i parametri, determinare le normali linee di base e le anomalie superficiali. Puoi creare allarmi di rilevamento delle anomalie basati sul valore previsto di un parametro. Ad esempio, è possibile monitorare le anomalie del parametro Object Lock Enabled Bytes (Byte con blocco oggetti abilitato) per rilevare una rimozione non autorizzata delle impostazioni di Blocco oggetti.
- Matematica dei parametri – permette di eseguire query di più parametri S3 Storage Lens e di utilizzare espressioni matematiche per creare nuove serie temporali in base a tali parametri. Ad esempio, è possibile creare un nuovo parametro per ottenere la dimensione media dell'oggetto dividendo `StorageBytes` per `ObjectCount`.

Per ulteriori informazioni sull'opzione di pubblicazione di CloudWatch per i parametri di S3 Storage Lens, consulta gli argomenti riportati di seguito.

Argomenti

- Dimensioni e parametri di S3 Storage Lens
- Abilitazione della pubblicazione di CloudWatch per S3 Storage Lens
- Utilizzo dei parametri di S3 Storage Lens su CloudWatch

Dimensioni e parametri di S3 Storage Lens

Per inviare parametri S3 Storage Lens a CloudWatch, è necessario abilitare l'opzione di pubblicazione di CloudWatch all'interno di Raccomandazioni e parametri avanzati di S3 Storage Lens. Una volta abilitati i parametri avanzati, è possibile utilizzare i pannelli di controllo CloudWatch per monitorare i parametri di S3 Storage Lens insieme ai parametri di altre applicazioni e crearne una visione unificata dello stato operativo. È possibile utilizzare le dimensioni per filtrare i parametri S3 Storage Lens su CloudWatch per organizzazione, account, bucket, classe di archiviazione, Regione e ID di configurazione dei parametri.

I parametri S3 Storage Lens sono pubblicati su CloudWatch nell'account che possiede la configurazione di S3 Storage Lens. Dopo aver abilitato l'opzione di pubblicazione CloudWatch all'interno dei parametri avanzati e suggerimenti, puoi accedere ai parametri a livello di

organizzazione, account e bucket in CloudWatch. I parametri a livello di prefissi non sono disponibili su CloudWatch.

> **Note**
>
> I parametri S3 Storage Lens sono parametri giornalieri e vengono pubblicati su CloudWatch una volta al giorno. Quando si interrogano i parametri S3 Storage Lens in CloudWatch, il periodo per la query deve essere di 1 giorno (86.400 secondi). Dopo che i parametri quotidiani di S3 Storage Lens sono visualizzati nel pannello di controllo di S3 Storage Lens nella console Amazon S3, possono essere necessarie alcune ore perché questi parametri siano visualizzati su CloudWatch. Quando abiliti per la prima volta l'opzione di pubblicazione di CloudWatch per i parametri S3 Storage Lens, possono essere necessarie fino a 24 ore prima che i tuoi parametri vengano pubblicati su CloudWatch.

Per ulteriori informazioni sulle dimensioni e sui parametri di S3 Storage Lens in CloudWatch, consulta gli argomenti riportati di seguito.

Argomenti

Parametri

I parametri di S3 Storage Lens sono disponibili come parametri all'interno di CloudWatch. I parametri di S3 Storage Lens sono pubblicati nello spazio dei nomi `AWS/S3/Storage-Lens`. Questo spazio dei nomi è solo per i parametri di S3 Storage Lens. I parametri bucket, richiesta e replica di Amazon S3 vengono pubblicati nello spazio dei nomi `AWS/S3`.

I parametri S3 Storage Lens sono pubblicati su CloudWatch nell'account che possiede la configurazione di S3 Storage Lens. Dopo aver abilitato l'opzione di pubblicazione CloudWatch all'interno dei parametri avanzati e suggerimenti, puoi accedere ai parametri a livello di organizzazione, account e bucket in CloudWatch. I parametri a livello di prefissi non sono disponibili su CloudWatch.

In S3 Storage Lens, i parametri vengono aggregati e memorizzati solo nella Regione di origine designata. I parametri di S3 Storage Lens vengono pubblicati su CloudWatch nella Regione originaria specificata nella configurazione di S3 Storage Lens.

Per un elenco completo dei parametri S3 Storage Lens, compreso l'elenco dei parametri disponibili su CloudWatch, vedi Glossario dei parametri di Amazon S3 Storage Lens.

> **Note**
>
> La statistica valida per i parametri di S3 Storage Lens in CloudWatch è Average. Per ulteriori informazioni sulle statistiche in CloudWatch, consulta CloudWatch statistics definitions (Definizioni delle statistiche di CloudWatch) nella Guida per l'utente di Amazon CloudWatch.

Granularità dei parametri di S3 Storage Lens su CloudWatch

S3 Storage Lens offre parametri sulla granularità dell'organizzazione, dell'account, del bucket e del prefisso. S3 Storage Lens pubblica su CloudWatch i parametri di organizzazione, account e S3 Storage Lens di livello bucket. I parametri a livello di prefissi S3 Storage Lens non sono disponibili su CloudWatch.

Per ulteriori informazioni sulla granularità dei parametri di S3 Storage Lens disponibili su CloudWatch, consulta il seguente elenco:

- Organizzazione – Parametri aggregati tra gli account membri della tua organizzazione. S3 Storage Lens pubblica i parametri degli account membri su CloudWatch nell'account di gestione.
  - Organizzazione e account – Parametri per gli account membri della tua organizzazione.
  - Organizzazione e account – Parametri per i bucket Amazon S3 buckets negli account membri della tua organizzazione.
- Account (Livello non di organizzazione): – Parametri aggregati tra i bucket del tuo account.
- Bucket (Livello non organizzativo) – Parametri di un bucket specifico. Su CloudWatch, S3 Storage Lens pubblica questi parametri sull'Account AWS che ha creato la configurazione di S3 Storage Lens. S3 Storage Lens pubblica questi parametri solo per configurazioni non organizzative.

Dimensioni

Quando S3 Storage Lens invia dati a CloudWatch, le dimensioni vengono collegate a ciascun parametro. Le dimensioni sono categorie che descrivono le caratteristiche dei parametri. Puoi utilizzare le dimensioni per filtrare i risultati restituiti da CloudWatch.

Ad esempio, tutti i parametri di S3 Storage Lens in CloudWatch hanno dimensione `configuration_id`. È possibile utilizzare questa dimensione per distinguere tra i parametri

associati con una configurazione specifica di S3 Storage Lens. L'`organization_id` identifica i parametri a livello organizzativo. Per ulteriori informazioni sulle dimensioni su CloudWatch, vedi Dimensions (Dimensioni) nella Guida per gli sviluppatori di CloudWatch.

Sono disponibili diverse dimensioni per i parametri S3 Storage Lens a seconda della granularità dei parametri. Ad esempio, è possibile utilizzare la dimensione `organization_id` per filtrare i parametri a livello di organizzazione in base all'ID AWS Organizations. Tuttavia, non è possibile utilizzare questa dimensione per parametri a livello di bucket e account. Per ulteriori informazioni, consulta Filtraggio dei parametri utilizzando le dimensioni.

Per vedere quali dimensioni sono disponibili per la configurazione di S3 Storage Lens, vedi la tabella seguente.

Dimensione	Descrizione	Bucket	Account	Organizzazione	Organizzazione e bucket	Organizzazione e account
`configuration_id`	Nome del pannello di controllo per la configurazione di S3 Storage Lens riportato nei parametri	✓	✓	✓	✓	✓
`metrics_version`	Versione dei parametri di S3 Storage Lens in uso. La versione dei parametri ha un valore fisso di `1.0`.	✓	✓	✓	✓	✓
`organization_id`	ID AWS Organizations dei parametri			✓	✓	✓
`aws_account_number`	Account AWS associato ai parametri	✓	✓		✓	✓
`aws_region`	Regione AWS dei parametri	✓	✓	✓	✓	✓
`bucket_name`	Nome del bucket S3 riportato nei parametri	✓	✓	✓	✓	✓
`storage_class`	Classe di archiviazione per il bucket riportato nei parametri	✓	✓	✓	✓	✓

Dimensione	Descrizione	Bucket	Account	Organizzazione	Organizzazione e bucket	Organizzazione e account
`record_type`	Granularità dei parametri: ORGANIZZAZIONE, ACCOUNT, BUCKET	BUCKET	ACCOUNT	ORGANIZZAZIONE	ORGANIZZAZIONE	ORGANIZZAZIONE

Abilitazione della pubblicazione di CloudWatch per S3 Storage Lens

Puoi pubblicare i parametri di S3 Storage Lens su Amazon CloudWatch per creare una visualizzazione unificata dello stato operativo nei pannelli di controllo di CloudWatch. È inoltre possibile utilizzare le funzioni di CloudWatch, come allarmi e azioni attivate, matematica dei parametri e rilevamento delle anomalie, per monitorare e intervenire sui parametri di S3 Storage Lens. Inoltre, le operazioni API di CloudWatch consentono alle applicazioni, inclusi i provider di terze parti, di accedere ai parametri di S3 Storage Lens. Per ulteriori informazioni sulle caratteristiche di CloudWatch, consulta la Guida per l'utente di Amazon CloudWatch.

I parametri S3 Storage Lens sono pubblicati su CloudWatch nell'account che possiede la configurazione di S3 Storage Lens. Dopo aver abilitato l'opzione di pubblicazione CloudWatch all'interno dei parametri avanzati e suggerimenti, puoi accedere ai parametri a livello di organizzazione, account e bucket in CloudWatch. I parametri a livello di prefissi non sono disponibili su CloudWatch.

Puoi abilitare il supporto di CloudWatch per configurazioni di pannello di controllo nuove o esistenti utilizzando la console S3, le REST API di Amazon S3, la AWS CLI e gli SDK AWS L'opzione di pubblicazione CloudWatch è disponibile per i pannelli di controllo aggiornati in base all'opzione S3 Storage Lens Advanced metrics and recommendations (Parametri avanzati e suggerimenti). Per i prezzi relativi all'opzione Advanced metrics and recommendations (Parametri avanzati e suggerimenti), consulta Prezzi di Amazon S3. Non vengono applicati ulteriori addebiti di pubblicazione dei parametri CloudWatch; tuttavia, sono applicabili altri costi di CloudWatch come pannelli di controllo, allarmi e chiamate API.

Per abilitare l'opzione di pubblicazione di CloudWatch per i parametri di S3 Storage Lens, consulta gli argomenti riportati di seguito.

> **Note**
>
> I parametri S3 Storage Lens sono parametri giornalieri e vengono pubblicati su CloudWatch una volta al giorno. Quando si interrogano i parametri S3 Storage Lens in CloudWatch, il periodo per la query deve essere di 1 giorno (86.400 secondi). Dopo che i parametri quotidiani di S3 Storage Lens sono visualizzati nel pannello di controllo di S3 Storage Lens nella console Amazon S3, possono essere necessarie alcune ore perché questi parametri siano visualizzati su CloudWatch. Quando abiliti per la prima volta l'opzione di pubblicazione di CloudWatch per i parametri S3 Storage Lens, possono essere necessarie fino a 24 ore prima che i tuoi parametri vengano pubblicati su CloudWatch.
> Attualmente, i parametri di S3 Storage Lens non possono essere utilizzati tramite flussi di CloudWatch.

Utilizzo della console S3

Quando viene aggiornato un pannello di controllo S3 Storage Lens, non è possibile modificare il nome del pannello di controllo o la regione di origine. Non è inoltre possibile cambiare l'ambito del pannello di controllo predefinito, che viene inserito nell'ambito dell'intera archiviazione dell'account.

Per aggiornare S3 Storage Lens per abilitare la pubblicazione CloudWatch

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli S3 Storage Lens, Dashboards.(Pannelli di controllo).
3. Scegli il pannello di controllo che desideri modificare, quindi seleziona Edit (Modifica).
4. Sotto Metrics selection (Selezione dei parametri), scegli Advanced metrics and recommendations (Raccomandazioni e parametri avanzati).

   Le raccomandazioni e i parametri avanzati sono dispobibili a un costo aggiuntivo. I suggerimenti e i parametri avanzati includono un periodo di 15 mesi per query di dati, parametri d'uso aggregati a livello di prefisso, parametri di attività aggregati per bucket, l'opzione pubblicazione CloudWatch e suggerimenti contestuali che consentono di ottimizzare i costi di archiviazione e applicare le best practice di protezione dei dati. Per ulteriori informazioni, consulta i Prezzi di Amazon S3.

5. Sotto Select Advanced metrics and recommendations features (Seleziona e parametri avanzati e funzioni di raccomandazione), seleziona CloudWatch publishing (Pubblicazione CloudWatch).

> **Important**
>
> Se la configurazione abilita l'aggregazione dei prefissi per i parametri di utilizzo, i parametri a livello di prefisso non verranno pubblicati su CloudWatch. In CloudWatch sono pubblicati solo i parametri di bucket, account e S3 Storage Lens a livello organizzativo.

6. Seleziona Salva modifiche.

Per creare un nuovo pannello di controllo di S3 Storage Lens che consente il supporto di CloudWatch

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Seleziona Crea pannello di controllo.
4. In General (Generale), definire le opzioni di configurazione seguenti:

   a. In Dashborard name (Nome pannello di controllo), inserisci il nome del pannello di controllo.

      I nomi del pannello di controllo devono contenere meno di 65 caratteri e non possono contenere caratteri speciali o spazi. Il nome del pannello di controllo dopo la creazione non potrà più essere modificato.

   b. Seleziona la Regione di origine del tuo pannello di controllo.

      I parametri per tutte le Regioni incluse nell'ambito di questo pannello di controllo vengono archiviati centralmente in questa Regione di origine designata. In CloudWatch, i parametri S3 Storage Lens sono disponibili anche nella Regione principale. Non è possibile modificare la regione di origine dopo aver creato il pannello di controllo.

5. (Facoltativo) Per aggiungere un tag, scegliere Add new tag (Aggiungi nuovo tag) e immettere la chiave e il valore del tag.

> **Note**
>
> Puoi aggiungere fino a 50 tag alla configurazione del pannello di controllo.

6. Definisci l'ambito della tua configurazione:

   a. Se stai creando una configurazione a livello di organizzazione, scegli gli account da includere nella configurazione: Include all accounts in your configuration (Includi tutti gli account nella configurazione) o Limit the scope to your signed-in account (Limita l'ambito al tuo account con accesso).

   > **Note**
   >
   > Quando si crea una configurazione a livello di organizzazione che include tutti gli account, è possibile includere o escludere solo regioni e non bucket.

   b. Seleziona le regioni e i bucket da includere in o escludere dalla configurazione del pannello di controllo eseguendo le seguenti operazioni:

   - Per includere tutte le Regioni, seleziona Include Regions and buckets (Includi Regioni e bucket).
   - Per includere Regioni specifiche, spunta Include all regions (Includi tutte le Regioni). Sotto Choose regions to include (Scegli le Regioni da includere), scegli le Regioni che desideri siano incluse nel pannello di controllo di S3 Storage Lens.
   - Per includere bucket specifici, spunta Include all buckets (Includi tutti i bucket). Sotto Choose buckets to include (Scegli i bucket da includere), scegli i bucket che desideri siano inclusi da S3 Storage Lens nel pannello di controllo.

   > **Note**
   >
   > Puoi scegliere fino a 50 bucket.

7. In Metrics selection (Selezione dei parametri), scegli Advanced metrics and recommendations (Parametri avanzati e suggerimenti).

   Per ulteriori informazioni sui prezzi avanzati e parametri avanzati, consulta Prezzi di Amazon S3.

8. In Advanced metrics and recommendations features (Parametri avanzati e funzioni di suggerimento), seleziona le opzioni da abilitare:

   - Advanced metrics (Parametri avanzati)
   - Pubblicazione CloudWatch

> **Important**
>
> Se abiliti l'aggregazione dei prefissi per la configurazione di S3 Storage Lens, i parametri a livello di prefisso non verranno pubblicati su CloudWatch. In CloudWatch sono pubblicati solo i parametri di bucket, account e S3 Storage Lens a livello organizzativo.

- Aggregazione di prefisso

> **Note**
>
> Per ulteriori informazioni sui parametri avanzati e sulle funzioni di suggerimento, consulta Selezione dei parametri.

9. Se hai abilitato Advanced metrics (Parametri avanzati), in Advanced metrics categories (Categorie parametri avanzati) seleziona le categorie che desideri visualizzare nel pannello di controllo di S3 Storage Lens:

   - Parametri delle attività
   - Detailed status code metrics (Parametri dettagliati codice di stato)
   - Advanced cost optimization metrics (Parametri avanzati ottimizzazione costi)
   - Advanced data protection metrics (Parametri avanzati protezione dati)

   Per ulteriori informazioni sulle categorie di parametri, consulta Categorie di parametri. Per un elenco completo di parametri, consulta Glossario dei parametri di Amazon S3 Storage Lens.

10. (Facoltativo) Configura l'esportazione dei parametri.

    Per ulteriori informazioni su come configurare l'esportazione dei parametri, consulta la sezione Creazione di un pannello di controllo di Amazon S3 Storage Lens.

11. Seleziona Crea pannello di controllo.

Utilizzo di AWS CLI

Il seguente esempio della AWS CLI abilita l'opzione di pubblicazione CloudWatch utilizzando una configurazione S3 Storage Lens a livello di organizzazione dei parametri avanzati e dei suggerimenti. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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.
    }
  }
}
```

Utilizzo dell'SDK AWS per 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();
        }
    }
}
```

Utilizzo di REST API

Per abilitare l'opzione di pubblicazione CloudWatch utilizzando la REST API Amazon S3, puoi utilizzare PutStorageLensConfiguration.

Fasi successive

Dopo aver abilitato l'opzione di pubblicazione di CloudWatch, puoi accedere ai parametri S3 Storage Lens in CloudWatch. Puoi anche sfruttare le funzionalità CloudWatch per monitorare e analizzare i dati del tuo S3 Storage Lens su CloudWatch. Per ulteriori informazioni, consulta i seguenti argomenti:

- Dimensioni e parametri di S3 Storage Lens
- Utilizzo dei parametri di S3 Storage Lens su CloudWatch

Utilizzo dei parametri di S3 Storage Lens su CloudWatch

Puoi pubblicare i parametri di S3 Storage Lens su Amazon CloudWatch per creare una visualizzazione unificata dello stato operativo nei pannelli di controllo di CloudWatch. È inoltre possibile utilizzare le funzioni di CloudWatch, come allarmi e azioni attivate, matematica dei parametri e rilevamento delle anomalie, per monitorare e intervenire sui parametri di S3 Storage Lens. Inoltre, le operazioni API di CloudWatch consentono alle applicazioni, inclusi i provider di terze parti, di accedere ai parametri di S3 Storage Lens. Per ulteriori informazioni sulle caratteristiche di CloudWatch, consulta la Guida per l'utente di Amazon CloudWatch.

Puoi abilitare l'opzione di pubblicazione CloudWatch per configurazioni di un pannello di controllo nuove o esistenti utilizzando la console Amazon S3, le REST API Amazon S3, la AWS CLI, e gli SDK AWS. L'opzione di pubblicazione CloudWatch è disponibile per i pannelli di controllo aggiornati in base all'opzione S3 Storage Lens Advanced metrics and recommendations (Parametri avanzati e suggerimenti). Per i prezzi relativi all'opzione Advanced metrics and recommendations (Parametri avanzati e suggerimenti), consulta Prezzi di Amazon S3. Non vengono applicati ulteriori addebiti di pubblicazione dei parametri CloudWatch; tuttavia, sono applicabili altri costi di CloudWatch come pannelli di controllo, allarmi e chiamate API. Per ulteriori informazioni, consulta Prezzi di Amazon CloudWatch.

I parametri S3 Storage Lens sono pubblicati su CloudWatch nell'account che possiede la configurazione di S3 Storage Lens. Dopo aver abilitato l'opzione di pubblicazione CloudWatch all'interno dei parametri avanzati e suggerimenti, puoi accedere ai parametri a livello di organizzazione, account e bucket in CloudWatch. I parametri a livello di prefissi non sono disponibili su CloudWatch.

> **Note**
>
> I parametri S3 Storage Lens sono parametri giornalieri e vengono pubblicati su CloudWatch una volta al giorno. Quando si interrogano i parametri S3 Storage Lens in CloudWatch, il periodo per la query deve essere di 1 giorno (86.400 secondi). Dopo che i parametri quotidiani di S3 Storage Lens sono visualizzati nel pannello di controllo di S3 Storage Lens nella console Amazon S3, possono essere necessarie alcune ore perché questi parametri siano visualizzati su CloudWatch. Quando abiliti per la prima volta l'opzione di pubblicazione di CloudWatch per i parametri S3 Storage Lens, possono essere necessarie fino a 24 ore prima che i tuoi parametri vengano pubblicati su CloudWatch.
> Attualmente, i parametri di S3 Storage Lens non possono essere utilizzati tramite flussi di CloudWatch.

Per ulteriori informazioni sull'utilizzo dei parametri S3 Storage Lens in CloudWatch, vedere i seguenti argomenti.

Argomenti

- Utilizzo dei pannelli di controllo CloudWatch
- Impostazione di allarmi, attivazione di azioni e utilizzo del rilevamento delle anomalie
- Filtraggio dei parametri utilizzando le dimensioni
- Calcolo di nuovi parametri con matematica dei parametri
- Utilizzo delle espressioni di ricerca nei grafici

Utilizzo dei pannelli di controllo CloudWatch

È possibile utilizzare i pannelli di controllo CloudWatch per monitorare i parametri di S3 Storage Lens insieme ai parametri di altre applicazioni e creare una visione unificata dello stato operativo. Un pannello di controllo è una home page personalizzabile nella console di CloudWatch che puoi utilizzare per monitorare le risorse in una visualizzazione singola.

CloudWatch ha un ampio controllo delle autorizzazioni che non supporta la limitazione dell'accesso a specifiche impostazioni di parametri o dimensioni. Gli utenti del tuo account o organizzazione che hanno accesso a CloudWatch avranno accesso ai parametri di tutte le configurazioni di S3 Storage Lens in cui è abilitata l'opzione di supporto CloudWatch. Non è possibile gestire autorizzazioni per pannelli di controllo specifici come è possibile in S3 Storage Lens. Per ulteriori informazioni sui

permessi di CloudWatch, consulta Managing access permissions to your CloudWatch resources (Gestione dei permessi di accesso alle risorse CloudWatch) nella Guida per l'utente di Amazon CloudWatch.

Per ulteriori informazioni sull'uso dei pannelli di controllo di CloudWatch e sulla configurazione delle autorizzazioni, consulta Using Amazon CloudWatch dashboards (Uso dei pannelli di controllo di Amazon CloudWatch) e Sharing CloudWatch dashboards (Condivisione dei pannelli di CloudWatch) nella Guida per l'utente di Amazon CloudWatch.

Impostazione di allarmi, attivazione di azioni e utilizzo del rilevamento delle anomalie

È possibile configurare gli allarmi CloudWatch che controllano i parametri di S3 Storage Lens in CloudWatch e intervengono quando viene violata una soglia. Ad esempio, è possibile configurare un allarme che invia una notifica di Amazon SNS quando il valore del parametro Incomplete Multipart Upload Bytes (Byte caricamenti in più parti incompleti) supera 1 GB per tre giorni consecutivi.

È inoltre possibile abilitare il rilevamento delle anomalie per analizzare continuamente i parametri di S3 Storage Lens, determinare le normali linee di base e le anomalie superficiali. Puoi creare allarmi di rilevamento delle anomalie basati sul valore previsto di un parametro. Ad esempio, è possibile monitorare le anomalie del parametro Object Lock Enabled Bytes (Byte con blocco oggetti abilitato) per rilevare una rimozione non autorizzata delle impostazioni di Blocco oggetti.

Per ulteriori informazioni ed esempi, consulta Using Amazon CloudWatch alarms (Uso degli allarmi Amazon CloudWatch) e Creating an alarm from a metric on a graph (Creazione di un allarme a partire da un parametro in un grafico) nella Guida per l'utente di Amazon CloudWatch.

Filtraggio dei parametri utilizzando le dimensioni

È possibile utilizzare le dimensioni per filtrare i parametri di S3 Storage Lens nella console CloudWatch. Ad esempio, è possibile filtrare per `configuration_id`, `aws_account_number`, `aws_region`, `bucket_name` e altri.

S3 Storage Lens supporta più configurazioni di pannello di controllo per account. Ciò significa che diverse configurazioni possono includere lo stesso bucket. Quando questi parametri vengono pubblicati su CloudWatch, il bucket avrà parametri duplicati all'interno di CloudWatch. Per visualizzare i parametri solo per una configurazione specifica di S3 Storage Lens in CloudWatch, è possibile utilizzare la dimensione `configuration_id`. Quando si filtra per `configuration_id`, vengono visualizzati solo i parametri associati alla configurazione identificata.

Per ulteriori informazioni sul filtro per ID di configurazione, consulta Ricerca dei parametri disponibili nella Guida per l'utente di Amazon CloudWatch.

Calcolo di nuovi parametri con matematica dei parametri

La matematica dei parametri permette di eseguire query di più parametri S3 Storage Lens e di utilizzare espressioni matematiche per creare nuove serie temporali in base a tali parametri. Ad esempio, è possibile creare un nuovo parametro per oggetti non crittografati sottraendo oggetti crittografati dal Conteggio oggetti. È inoltre possibile creare un parametro per ottenere la dimensione media dell'oggetto dividendo `StorageBytes` per `ObjectCount`, o i byte numerici a cui si accede in un giorno dividendo `BytesDownloaded` per `StorageBytes`.

Per ulteriori informazioni, consulta Using metric math (Utilizzo di matematica dei parametri) nella Guida per l'utente di Amazon CloudWatch.

Utilizzo delle espressioni di ricerca nei grafici

Con i parametri di S3 Storage Lens, puoi creare un'espressione di ricerca. Ad esempio, puoi creare un'espressione di ricerca per tutti i parametri denominati IncompleteMultipartUploadStorageBytes e aggiungere `SUM` all'espressione. Con questa espressione di ricerca, è possibile visualizzare il valore del parametro Incomplete Multipart Upload Bytes (Byte caricamenti in più parti incompleti) per tutte le dimensioni dell'archiviazione in un unico parametro.

Questo esempio mostra la sintassi da utilizzare per creare un'espressione di ricerca per tutti i parametri denominati IncompleteMultipartUploadStorageBytes.

```
SUM(SEARCH('{AWS/S3/Storage-
Lens,aws_account_number,aws_region,configuration_id,metrics_version,record_type,storage_class}
 MetricName="IncompleteMultipartUploadStorageBytes"', 'Average',86400))
```

Per ulteriori informazioni su questa sintassi, vedi CloudWarch search expression syntax (Sintassi dell'espressione di ricerca di CloudWatch) nella Guida per l'utente di Amazon CloudWatch. Per creare un grafico CloudWatch con un'espressione di ricerca, vedi Creating a CloudWatch graph with a search expression (Creazione di un grafico CloudWatch con un'espressione di ricerca) nella Guida per l'utente di Amazon CloudWatch.

Casi d'uso relativi ai parametri di Amazon S3 Storage Lens

Puoi utilizzare il pannello di controllo di Amazon S3 Storage Lens per visualizzare informazioni dettagliate e tendenze, contrassegnare le anomalie e ricevere suggerimenti. I parametri di S3 Storage Lens sono organizzati in categorie conformi ai principali casi di utilizzo. Puoi utilizzare questi parametri per effettuare le seguenti operazioni:

- Individuare le opportunità di ottimizzazione dei costi
- Applicare le best practice per la protezione dei dati
- Applica le best practice per la gestione degli accessi
- Migliorare le prestazioni dei carichi di lavoro delle applicazioni

Ad esempio, con i parametri relativi alla ottimizzazione dei costi, è possibile individuare le opportunità di riduzione dei costi di archiviazione di Amazon S3. Puoi individuare i bucket con caricamenti in più parti che risalgono a più di 7 giorni fa o i bucket che accumulano versioni non correnti.

Allo stesso modo, puoi utilizzare i parametri relativi alla protezione dei dati per individuare i bucket non conformi alle best practice di protezione dei dati all'interno dell'organizzazione. Ad esempio, puoi individuare i bucket che non utilizzano chiavi AWS Key Management Service (SSE-KMS) per la crittografia predefinita o che non hanno la funzionalità S3 di controllo delle versioni abilitata.

Con i parametri relativi alla gestione degli accessi di S3 Storage Lens, puoi identificare le impostazioni della caratteristica S3 Object Ownership per i bucket in modo da poter eseguire la migrazione delle autorizzazioni delle liste di controllo degli accessi (ACL) alle policy di bucket e disabilitare le ACL.

Se hai abilitato l'opzione Advanced metrics (Parametri avanzati) di S3 Storage Lens, puoi utilizzare i parametri dei codici di stato dettagliati per ottenere i numeri delle richieste riuscite o non riuscite per la risoluzione dei problemi relativi ad accessi e prestazioni.

Con i parametri avanzati, puoi anche accedere a parametri aggiuntivi relativi all'ottimizzazione dei costi e alla protezione dei dati che puoi utilizzare per individuare le opportunità per ridurre ulteriormente i costi complessivi dell'archiviazione S3 e allinearti meglio alle best practice per la protezione dei dati. Ad esempio, i parametri avanzati relativi all'ottimizzazione dei costi includono i conteggi delle regole del ciclo di vita che puoi utilizzare per identificare i bucket senza regole del ciclo di vita per far scadere i caricamenti in più parti incompleti che risalgono a più di 7 giorni fa. I parametri avanzati relativi alla protezione dei dati includono il conteggio delle regole di replica.

Per ulteriori informazioni sulle categorie di parametri, consulta Categorie di parametri. Per un elenco completo dei parametri di S3 Storage Lens, consulta Glossario dei parametri di Amazon S3 Storage Lens.

Argomenti

- Utilizzo di Amazon S3 Storage Lens per ottimizzare i costi di archiviazione

- Utilizzo di S3 Storage Lens per proteggere i tuoi dati
- Utilizzo di S3 Storage Lens per controllare le impostazioni di Object Ownership
- Utilizzo dei parametri di S3 Storage Lens per migliorare le prestazioni

Utilizzo di Amazon S3 Storage Lens per ottimizzare i costi di archiviazione

Puoi utilizzare i parametri di ottimizzazione dei costi di S3 Storage Lens per ridurre il costo complessivo dell'archiviazione S3. I parametri di ottimizzazione dei costi possono aiutarti a confermare di aver configurato Amazon S3 in modo conveniente e in modo conforme alle best practice. Ad esempio, è possibile individuare le seguenti opportunità di ottimizzazione dei costi:

- Bucket con caricamenti in più parti incompleti più vecchi di 7 giorni
- Bucket che accumulano numerose versioni non correnti
- Bucket che non hanno regole del ciclo di vita per l'interruzione dei caricamenti in più parti incompleti
- Bucket che non dispongono di regole del ciclo di vita per far scadere gli oggetti delle versioni non correnti
- Bucket che non dispongono di regole del ciclo di vita per trasferire gli oggetti a una classe di archiviazione diversa

È quindi possibile utilizzare questi dati per aggiungere ulteriori regole del ciclo di vita ai bucket.

Di seguito sono riportati esempi che mostrano come utilizzare i parametri di ottimizzazione dei costi nel pannello di controllo di S3 Storage Lens per ottimizzare i costi di archiviazione.

Argomenti

- Identificare i bucket S3 più grandi
- Scoperta dei bucket Amazon S3 freddi
- Individuazione di caricamenti in più parti incompleti
- Riduzione del numero di versioni non correnti conservate
- Identificare i bucket senza regole del ciclo di vita ed esaminare i conteggi delle regole del ciclo di vita

Identificare i bucket S3 più grandi

L'archiviazione degli oggetti nei bucket S3 è a pagamento. La tariffa che ti viene addebitata dipende dalle dimensioni degli oggetti, dalla durata di archiviazione degli oggetti e dalle relative classi di archiviazione. Con S3 Storage Lens, ottieni una vista centralizzata di tutti i bucket nel tuo account. Per visualizzare tutti i bucket in tutti gli account della tua organizzazione, puoi configurare un pannello di controllo di S3 Storage Lens a livello di AWS Organizations. Da questa vista del pannello di controllo è possibile identificare i bucket più grandi.

Fase 1: identificare i bucket più grandi

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Pannelli di controllo, scegli il pannello di controllo che desideri visualizzare.

   Quando il pannello di controllo si apre, puoi vedere l'ultima data in cui S3 Storage Lens ha raccolto i parametri. Il pannello di controllo viene sempre caricato alla data più recente per la quale sono disponibili i parametri.
4. Per visualizzare una classifica dei bucket più grandi in base al parametro Total storage (Archiviazione totale) per un intervallo di date selezionato, scorri verso il basso fino alla sezione Top N overview for date (Panoramica primi N per [data]).

   È possibile modificare l'ordinamento per mostrare i bucket più piccoli. Puoi anche modificare la selezione in Metric (Parametro) per classificare i tuoi bucket in base a uno qualsiasi dei parametri disponibili. La sezione Top N overview for date (Panoramica primi N per [data]) mostra anche la variazione percentuale rispetto al giorno o alla settimana precedente e un grafico sparkline per visualizzare la tendenza. Questa tendenza è valida per 14 giorni per i parametri gratuiti e per 30 giorni per i parametri e i suggerimenti avanzati.

   > **Note**
   >
   > Con i parametri avanzati e i suggerimenti di S3 Storage Lens, i parametri sono disponibili per le query per 15 mesi. Per ulteriori informazioni, consulta Selezione dei parametri.
5. Per informazioni più dettagliate sui bucket, scorri fino alla parte superiore della pagina, quindi scegli la scheda Bucket.

Nella scheda Bucket è possibile visualizzare dettagli quali il tasso di crescita recente, la dimensione media dell'oggetto, i prefissi più grandi e il numero di oggetti.

Fase 2: accedere ai bucket e analizzare

Per i bucket S3 più grandi, è quindi possibile passare a ciascun bucket all'interno della console S3 per analizzare i relativi oggetti e il carico di lavoro associato o per identificarne i proprietari interni. Puoi contattare i proprietari del bucket per scoprire se questa crescita è prevista o se necessita di ulteriore monitoraggio e controllo.

Scoperta dei bucket Amazon S3 freddi

Se hai l'opzione Parametri avanzati di S3 Storage Lensabilitata, puoi utilizzare i parametri delle attività per capire quanto sono freddi i tuoi bucket S3. Un bucket "freddo" è un bucket il cui spazio di archiviazione non è più utilizzato o è utilizzato molto raramente. Questa mancanza di attività indica in genere che gli oggetti del bucket non vengono utilizzati frequentemente.

Parametri di attività, quali Richieste GET e Byte di download, indicano la frequenza di accesso ai tuoi bucket ogni giorno. Per comprendere la coerenza del modello di accesso e individuare i bucket a cui non si accede più, è possibile seguire l'andamento di questi dati per diversi mesi. Il parametro Tasso di recupero, che viene calcolato come Byte di download/Spazio di archiviazione totale, indica la proporzione di spazio di archiviazione in un bucket a cui si accede quotidianamente.

> **Note**
>
> I byte di download vengono duplicati nei casi in cui lo stesso oggetto venga scaricato più volte durante il giorno.

Prerequisito

Per visualizzare i parametri relativi alle attività nel pannello di controllo di S3 Storage Lens, devi abilitare l'opzione S3 Storage Lens Advanced metrics and recommendations (Parametri e suggerimenti avanzati) e quindi selezionare Activity metrics (Parametri attività). Per ulteriori informazioni, consulta Creazione e aggiornamento dei pannelli di controllo di Amazon S3 Storage Lens.

Fase 1: identificare i bucket attivi

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Pannelli di controllo, scegli il pannello di controllo che desideri visualizzare.
4. Seleziona la scheda Bucket e scorri verso il basso fino alla sezione grafici Bubble analysis by buckets for date (Analisi a bolle per bucket per [data]).

   Nella sezione Bubble analysis by buckets for date (Analisi a bolle per bucket per [data]), è possibile tracciare i bucket su più dimensioni utilizzando tre parametri per rappresentare l'asse X, l'asse Y e la dimensione della bolla.
5. Per trovare i bucket "freddi", per l'asse X, l'asse Y e la dimensione, scegli i parametri Total storage (Archiviazione totale), % retrieval rate (% frequenza recupero) e Average object size (Dimensione media oggetto).
6. Nella sezione Bubble analysis by buckets for date (Analisi a bolle per bucket per [data]), individua tutti i bucket con frequenze di recupero pari a zero (o vicino a zero) e una dimensione di archiviazione relativa maggiore e quindi scegli la bolla che rappresenta il bucket.

   Apparirà un riquadro con le scelte per visualizzare informazioni dettagliate più granulari. Completa una delle seguenti operazioni:

   a. Per aggiornare la scheda Bucket in modo da visualizzare i parametri solo per il bucket selezionato, scegli Drill down (Esegui drill-down), quindi scegli Apply (Applica).
   b. Per aggregare i dati a livello di bucket per account, Regione AWS, classe di archiviazione o bucket, scegli Analyze by (Analizza per), quindi seleziona un valore in Dimension (Dimensione). Ad esempio, per eseguire l'aggregazione per classe di archiviazione, scegli Storage class (Classe di archiviazione) in Dimension (Dimensione).

   Per trovare i bucket che si sono raffreddati, esegui un'analisi delle bolle utilizzando i parametri Archiviazione totale, % tasso di recupero e Dimensione media degli oggetti. Cerca tutti i bucket con tasso di recupero pari a zero (o vicino a zero) e una dimensione di archiviazione relativa maggiore.

   La scheda Bucket del pannello di controllo viene aggiornata con i dati per l'aggregazione o il filtro selezionato. Se hai effettuato l'aggregazione per classe di archiviazione o un'altra dimensione,

la nuova scheda, ad esempio la scheda Storage class (Classe di archiviazione), si apre nel pannello di controllo.

Fase 2: analizzare i bucket freddi

Da qui è possibile identificare i proprietari del bucket freddi nel tuo account o nella tua organizzazione e scoprire se lo spazio di archiviazione è ancora necessario. È quindi possibile ottimizzare i costi impostando le configurazioni della scadenza del ciclo di vita per i bucket o archiviando i dati in una delle classi di archiviazione Amazon S3 Glacier.

Per evitare il problema dei bucket freddi, è possibile eseguire una transizione automatica dei dati utilizzando le configurazioni del ciclo di vita S3 per i tuoi bucket oppure puoi abilitare l'archiviazione automatica con Piano intelligente Amazon S3.

È inoltre possibile utilizzare la fase 1 per identificare i bucket "caldi". Quindi, puoi assicurarti che questi bucket utilizzino la classe di archiviazione S3 corretta per garantire che soddisfino le loro richieste nel modo più efficace in termini di prestazioni e costi.

Individuazione di caricamenti in più parti incompleti

Puoi utilizzare i caricamenti in più parti per caricare oggetti di grandi dimensioni (fino a 5 TB) come set di parti per migliorare la velocità di trasmissione effettiva ed eseguire più rapidamente il ripristino in caso di problemi di rete. Nei casi in cui il processo di caricamento in più parti non venga portato a termine, le parti incomplete rimangono nel bucket (in uno stato inutilizzabile). Queste parti incomplete comportano costi di archiviazione fino al termine del processo di caricamento o fino alla rimozione delle parti incomplete. Per ulteriori informazioni, consulta Caricamento e copia di oggetti utilizzando il caricamento in più parti.

Con S3 Storage Lens, puoi identificare il numero di byte di un caricamento in più parti incompleto nel tuo account o nell'intera organizzazione, compresi i caricamenti in più parti incompleti che risalgono a più di 7 giorni. Per un elenco completo dei parametri relativi ai caricamenti in più parti incompleti, consulta Glossario dei parametri di Amazon S3 Storage Lens.

Come best practice, consigliamo di configurare le regole del ciclo di vita per far scadere i caricamenti in più parti incompleti più vecchi di un determinato numero di giorni. Quando crei la regola del ciclo di vita per far scadere i caricamenti in più parti incompleti, consigliamo il valore di 7 giorni come buon punto di partenza.

Fase 1: esaminare le tendenze generali relative ai caricamenti incompleti in più parti

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Pannelli di controllo, scegli il pannello di controllo che desideri visualizzare.
4. Nella sezione Snapshot for date (Snapshot per [data]), in Metrics categories (Categorie parametri), scegli Cost optimization (Ottimizzazione costi).

   La sezione Snapshot for date (Snapshot per [data]) viene aggiornata per visualizzare i parametri Cost optimization (Ottimizzazione costi), che includono Incomplete multipart upload bytes greater than 7 days old (Byte caricamenti in più parti incompleti risalenti a più di 7 giorni).

   In tutti i grafici del pannello di controllo di S3 Storage Lens, puoi vedere i parametri relativi ai caricamenti in più parti incompleti. Puoi utilizzare questi parametri per valutare ulteriormente l'impatto dei byte dei caricamenti in più parti incompleti nell'archiviazione, incluso il loro contributo alle tendenze generali di crescita. Puoi anche eseguire il drill-down sui livelli di aggregazione in dettaglio utilizzando le schede Account, Regione AWS, Bucket o Storage class (Classe di archiviazione) per un'analisi più approfondita dei tuoi dati. Per un esempio, consulta Scoperta dei bucket Amazon S3 freddi.

Fase 2: identificare i bucket con i byte di caricamento in più parti più incompleti, ma che non dispongono di regole del ciclo di vita per interrompere i caricamenti in più parti incompleti

Prerequisito

Per visualizzare il parametro Abort incomplete multipart upload lifecycle rule count (Conteggio regole ciclo di vita interruzione caricamenti in più parti incompleti) nel pannello di controllo di S3 Storage Lens, devi abilitare l'opzione S3 Storage Lens Advanced metrics and recommendations (Parametri e suggerimenti avanzati) e quindi selezionare Advanced cost optimization metrics (Parametri avanzati ottimizzazione costi). Per ulteriori informazioni, consulta Creazione e aggiornamento dei pannelli di controllo di Amazon S3 Storage Lens.

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Pannelli di controllo, scegli il pannello di controllo che desideri visualizzare.

4. Per identificare i bucket specifici che accumulano caricamenti in più parti incompleti risalenti a più di 7 giorni, vai alla sezione Top N overview for date (Panoramica primi N per [data]).

   Per impostazione predefinita, la sezione Top N overview for date (Panoramica primi N per [data]) mostra i parametri per i primi 3 bucket. È possibile aumentare o diminuire il numero di bucket nel campo Top N (Primi N). La sezione Top N overview for date (Panoramica primi N per [data]) mostra anche la variazione percentuale rispetto al giorno o alla settimana precedente e un grafico sparkline per visualizzare la tendenza. Questa tendenza è valida per 14 giorni per i parametri gratuiti e per 30 giorni per i parametri e i suggerimenti avanzati.

   > **Note**
   >
   > Con i parametri avanzati e i suggerimenti di S3 Storage Lens, i parametri sono disponibili per le query per 15 mesi. Per ulteriori informazioni, consulta Selezione dei parametri.

5. In Metric (Parametro), scegli Incomplete multipart upload bytes greater than 7 days old (Byte caricamenti in più parti incompleti risalenti a più di 7 giorni) nella categoria Cost optimization (Ottimizzazione costi).

   Nella sezione Top number buckets (Primi [numero] bucket), puoi visualizzare i bucket con i byte di archiviazione per caricamenti in più parti incompleti che risalgono a più di 7 giorni.

6. Per visualizzare i parametri più dettagliati a livello di bucket per i caricamenti in più parti incompleti, scorri fino alla parte superiore della pagina, quindi scegli la scheda Bucket.

7. Scorri fino alla sezione Buckets (Bucket). In Metrics categories (Categorie parametri), seleziona Cost optimization (Ottimizzazione dei costi). Quindi deseleziona Summary (Riepilogo).

   L'elenco Buckets (Bucket) viene aggiornato per visualizzare tutti i parametri di ottimizzazione dei costi disponibili per i bucket visualizzati.

8. Per filtrare l'elenco Buckets (Bucket) in modo da visualizzare solo i parametri relativi all'ottimizzazione dei costi, scegli l'icona delle preferenze ().

9. Deseleziona tutti i parametri per l'ottimizzazione dei costi e lascia selezionati solo i parametri Incomplete multipart upload bytes greater than 7 days old (Byte caricamenti in più parti incompleti risalenti a più di 7 giorni) e Abort incomplete multipart upload lifecycle rule count (Conteggio regole ciclo di vita interruzione caricamenti in più parti incompleti).

10. (Facoltativo) In Page size (Dimensioni pagina), scegli il numero di bucket da visualizzare nell'elenco.

11. Scegli Confirm (Conferma).

    L'elenco Buckets (Bucket) viene aggiornato con i parametri a livello di bucket per i conteggi dei caricamenti in più parti incompleti e per le regole del ciclo di vita. Puoi utilizzare questi dati per identificare i bucket con i byte di caricamenti in più parti più incompleti che risalgono a più di 7 giorni e che non presentano regole del ciclo di vita per interrompere i caricamenti in più parti incompleti. Quindi, puoi passare a questi bucket nella console S3 e aggiungere regole del ciclo di vita per eliminare i caricamenti in più parti incompleti abbandonati.

Fase 3: aggiungere una regola del ciclo di vita per eliminare i caricamenti in più parti incompleti dopo 7 giorni

Per gestire automaticamente i caricamenti in più parti incompleti, è possibile utilizzare la console S3 per creare una configurazione del ciclo di vita per far scadere byte di caricamenti in più parti incompleti da un bucket dopo un determinato numero di giorni. Per ulteriori informazioni, consulta Configurazione del ciclo di vita del bucket per l'eliminazione dei caricamenti in più parti incompleti.

Riduzione del numero di versioni non correnti conservate

Se attivata, la funzionalità S3 di controllo delle versioni conserva più versioni dello stesso oggetto; tali versioni possono essere utilizzate per recuperare rapidamente i dati nel caso in cui un oggetto venga eliminato o sovrascritto accidentalmente. Se hai abilitato funzionalità S3 di controllo delle versioni senza configurare le regole del ciclo di vita per la transizione o la scadenza delle versioni non correnti, può accumularsi un gran numero di versioni precedenti non correnti, con ripercussioni sui costi di archiviazione. Per ulteriori informazioni, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Fase 1: identificare i bucket con il maggior numero di versioni di oggetti non correnti

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Pannelli di controllo, scegli il pannello di controllo che desideri visualizzare.
4. Nella sezione Snapshot for date (Snapshot per [data]), in Metric categories (Categorie parametri), scegli Cost optimization (Ottimizzazione costi).

   La sezione Snapshot for date (Snapshot per [data]) viene aggiornata per visualizzare i parametri Cost optimization (Ottimizzazione costi), che includono il parametro % noncurrent version bytes

(% byte versioni non correnti). Il parametro % noncurrent version bytes (% byte versioni non correnti) rappresenta la proporzione dei byte di archiviazione totali attribuita alle versioni non correnti nell'ambito del pannello di controllo e per la data selezionata.

> **Note**
>
> Se il valore del parametro % noncurrent version bytes (% byte versioni non correnti) è maggiore del 10% dell'archiviazione a livello di account, ciò può indicare che stai archiviando troppe versioni di oggetti.

5. Per identificare i bucket specifici che accumulano un numero elevato di versioni non correnti:

   a. Scorri verso il basso fino alla sezione Top N overview for date (Panoramica primi N per [data]). In Top N (Primi N), inserisci il numero di bucket per i quali desideri visualizzare i dati.

   b. In Metric (Parametro), scegli % noncurrent version bytes (% byte versioni non correnti).

      In Top number buckets (Primi [numero] bucket), puoi visualizzare i bucket (per il numero specificato) con il valore più alto del parametro % noncurrent version bytes (% byte versioni non correnti). La sezione Top N overview for date (Panoramica primi N per [data]) mostra anche la variazione percentuale rispetto al giorno o alla settimana precedente e un grafico sparkline per visualizzare la tendenza. Questa tendenza è valida per 14 giorni per i parametri gratuiti e per 30 giorni per i parametri e i suggerimenti avanzati.

      > **Note**
      >
      > Con i parametri avanzati e i suggerimenti di S3 Storage Lens, i parametri sono disponibili per le query per 15 mesi. Per ulteriori informazioni, consulta Selezione dei parametri.

   c. Per visualizzare i parametri più dettagliati a livello di bucket per le versioni di oggetti non correnti, scorri fino alla parte superiore della pagina, quindi scegli la scheda Bucket.

      In qualsiasi grafico o visualizzazione del pannello di controllo di S3 Storage Lens, puoi eseguire il drill-down dei livelli di aggregazione maggiormente in dettaglio utilizzando le schede Account, Regione AWS, Storage class (Classe di archiviazione) o Bucket. Per un esempio, consulta Scoperta dei bucket Amazon S3 freddi.

   d. Nella sezione Buckets (Bucket), in Metric categories (Categorie parametri), seleziona Cost optimization (Ottimizzazione dei costi). Quindi seleziona Summary (Riepilogo).

A questo punto puoi visualizzare il parametro % noncurrent version bytes (% byte versioni non correnti), insieme ad altri parametri relativi alle versioni non correnti.

Fase 2: identificare i bucket privi di regole del ciclo di vita di transizione e scadenza per la gestione delle versioni non correnti

Prerequisito

Per visualizzare i parametri Noncurrent version transition lifecycle rule count (Conteggio regole ciclo di vita transizioni versione non corrente) e Noncurrent version expiration lifecycle rule count (Conteggio regole ciclo di vita scadenza versione non corrente) nel pannello di controllo di S3 Storage Lens, devi abilitare l'opzione S3 Storage Lens Advanced metrics and recommendations (Parametri e suggerimenti avanzati) e quindi selezionare Advanced cost optimization metrics (Parametri avanzati ottimizzazione costi). Per ulteriori informazioni, consulta Creazione e aggiornamento dei pannelli di controllo di Amazon S3 Storage Lens.

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Pannelli di controllo, scegli il pannello di controllo che desideri visualizzare.
4. Nel pannello di controllo di Storage Lens, scegli la scheda Bucket.
5. Scorri fino alla sezione Buckets (Bucket). In Metrics categories (Categorie parametri), seleziona Cost optimization (Ottimizzazione dei costi). Quindi deseleziona Summary (Riepilogo).

   L'elenco Buckets (Bucket) viene aggiornato per visualizzare tutti i parametri di ottimizzazione dei costi disponibili per i bucket visualizzati.
6. Per filtrare l'elenco Buckets (Bucket) in modo da visualizzare solo i parametri relativi all'ottimizzazione dei costi, scegli l'icona delle preferenze ().
7. Deseleziona tutti i parametri di ottimizzazione dei costi finché non rimangono selezionati solo i seguenti parametri:

   - % noncurrent version bytes (% byte versioni non correnti)
   - Noncurrent version transition lifecycle rule count (Conteggio regole ciclo di vita transizione versioni non correnti)

- Noncurrent version expiration lifecycle rule count (Conteggio regole ciclo di vita scadenza versioni non correnti)

8. (Facoltativo) In Page size (Dimensioni pagina), scegli il numero di bucket da visualizzare nell'elenco.
9. Scegli Confirm (Conferma).

   L'elenco Buckets (Bucket) viene aggiornato con i parametri relativi ai byte di versioni non correnti e ai conteggi delle regole del ciclo di vita delle versioni non correnti. È possibile utilizzare questi dati per identificare i bucket che hanno un'alta percentuale di byte di versioni non correnti, ma sono privi di regole del ciclo di vita di transizione e scadenza. Quindi, puoi accedere a questi bucket nella console S3 e aggiungervi regole del ciclo di vita.

Fase 3: aggiungere regole del ciclo di vita per eseguire la transizione o la scadenza delle versioni degli oggetti non correnti

Dopo aver determinato quali bucket richiedono ulteriori indagini, puoi passare ai bucket all'interno della console S3 e aggiungere una regola del ciclo di vita per far scadere le versioni non correnti dopo un numero specificato di giorni. In alternativa, per ridurre i costi pur mantenendo le versioni non correnti, puoi configurare una regola del ciclo di vita per la transizione delle versioni non correnti a una delle classi di archiviazione Amazon S3 Glacier. Per ulteriori informazioni, consulta Esempio 6: specifica di una regola del ciclo di vita per un bucket che supporta la funzione Controllo delle versioni.

Identificare i bucket senza regole del ciclo di vita ed esaminare i conteggi delle regole del ciclo di vita

S3 Storage Lens fornisce parametri relativi al numero di regole del ciclo di vita S3 che puoi utilizzare per identificare i bucket senza regole del ciclo di vita. Per trovare i bucket senza regole del ciclo di vita, puoi utilizzare il parametro Total buckets without lifecycle rules (Totale bucket senza regole ciclo di vita). Un bucket senza una configurazione S3 del ciclo di vita potrebbe disporre di un'archiviazione non più necessaria o che può essere trasferita a una classe di archiviazione a basso costo. Puoi anche utilizzare i parametri relativi al conteggio delle regole del ciclo di vita per identificare i bucket senza tipi specifici di regole del ciclo di vita, come le regole di scadenza o di transizione.

Prerequisito

Per visualizzare i parametri relativi ai conteggi delle regole ciclo di vita e il parametro Total buckets without lifecycle rules (Totale bucket senza regole ciclo di vita) nel pannello di controllo di S3 Storage Lens, devi abilitare l'opzione S3 Storage Lens Advanced metrics and recommendations (Parametri e suggerimenti avanzati) e quindi selezionare Advanced cost optimization metrics (Parametri avanzati

ottimizzazione costi). Per ulteriori informazioni, consulta Creazione e aggiornamento dei pannelli di controllo di Amazon S3 Storage Lens.

Fase 1: identificare i bucket senza regole del ciclo di vita

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Pannelli di controllo, scegli il pannello di controllo che desideri visualizzare.
4. Per identificare i bucket specifici senza regole del ciclo di vita, scorri verso il basso fino alla sezione Top N overview for date (Panoramica primi N per [data]).

   Per impostazione predefinita, la sezione Top N overview for date (Panoramica primi N per [data]) mostra i parametri per i primi 3 bucket. Nel campo Top N (Primi N) è possibile aumentare il numero di bucket. La sezione Top N overview for date (Panoramica primi N per [data]) mostra anche la variazione percentuale rispetto al giorno o alla settimana precedente e un grafico sparkline per visualizzare la tendenza. Questa tendenza è valida per 14 giorni per i parametri gratuiti e per 30 giorni per i parametri e i suggerimenti avanzati.

   > **Note**
   >
   > Con i parametri avanzati e i suggerimenti di S3 Storage Lens, i parametri sono disponibili per le query per 15 mesi. Per ulteriori informazioni, consulta Selezione dei parametri.

5. In Metric (Parametro), scegli Total buckets without lifecycle rules (Totale bucket senza regole ciclo di vita) nella categoria Cost optimization (Ottimizzazione costi).
6. Esamina i seguenti dati per il parametro Total buckets without lifecycle rules (Totale bucket senza regole ciclo di vita):

   - Top number accounts (Primi [numero] account): visualizza gli account con il maggior numero di bucket senza regole del ciclo di vita.
   - Top number Regions (Prime [numero] regioni): visualizza un'analisi dettagliata dei bucket senza regole del ciclo di vita per regione.
   - Top number buckets (Primi [numero] bucket]: visualizza i bucket senza regole del ciclo di vita.

   In qualsiasi grafico o visualizzazione del pannello di controllo di S3 Storage Lens, puoi eseguire il drill-down dei livelli di aggregazione maggiormente in dettaglio utilizzando le schede Account,

Regione AWS, Storage class (Classe di archiviazione) o Bucket. Per un esempio, consulta Scoperta dei bucket Amazon S3 freddi.

Dopo aver identificato i bucket senza regole del ciclo di vita, puoi anche esaminare i conteggi specifici delle regole del ciclo di vita per i tuoi bucket.

Fase 2: rivedere i conteggi delle regole del ciclo di vita per i bucket

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Pannelli di controllo, scegli il pannello di controllo che desideri visualizzare.
4. Nel pannello di controllo di S3 Storage Lens, scegli la scheda Bucket.
5. Scorri fino alla sezione Buckets (Bucket). In Metrics categories (Categorie parametri), seleziona Cost optimization (Ottimizzazione dei costi). Quindi deseleziona Summary (Riepilogo).

   L'elenco Buckets (Bucket) viene aggiornato per visualizzare tutti i parametri di ottimizzazione dei costi disponibili per i bucket visualizzati.
6. Per filtrare l'elenco Buckets (Bucket) in modo da visualizzare solo i parametri relativi all'ottimizzazione dei costi, scegli l'icona delle preferenze ().
7. Deseleziona tutti i parametri di ottimizzazione dei costi finché non rimangono selezionati solo i seguenti parametri:

   - Transition lifecycle rule count (Conteggio regole ciclo di vita transizione)
   - Expiration lifecycle rule count (Conteggio regole ciclo di vita scadenza)
   - Noncurrent version transition lifecycle rule count (Conteggio regole ciclo di vita transizione versioni non correnti)
   - Noncurrent version expiration lifecycle rule count (Conteggio regole ciclo di vita scadenza versioni non correnti)
   - Abort incomplete multipart upload lifecycle rule count (Conteggio regole ciclo di vita interruzione caricamenti in più parti incompleti)
   - Total lifecycle rule count (Conteggio totale regole ciclo di vita)
8. (Facoltativo) In Page size (Dimensioni pagina), scegli il numero di bucket da visualizzare nell'elenco.

9. Scegli Confirm (Conferma).

   L'elenco Buckets (Bucket) viene aggiornato con i parametri relativi al conteggio delle regole del ciclo di vita per i bucket. È possibile utilizzare questi dati per identificare i bucket senza regole del ciclo di vita o i bucket a cui mancano tipi specifici di regole del ciclo di vita, ad esempio regole di scadenza o di transizione. Quindi, puoi accedere a questi bucket nella console S3 e aggiungervi regole del ciclo di vita.

Fase 3: aggiungere regole del ciclo di vita

Dopo aver identificato i bucket privi di regole del ciclo di vita, puoi aggiungere tali regole. Per ulteriori informazioni, consulta Impostazione di una configurazione del ciclo di vita su un bucket e Esempi di configurazione del ciclo di vita S3.

Utilizzo di S3 Storage Lens per proteggere i tuoi dati

Puoi utilizzare i parametri per la protezione dei dati di Amazon S3 Storage Lens per identificare i bucket in cui non sono state applicate le best practice per la protezione dei dati. Puoi utilizzare questi parametri per definire e applicare impostazioni standard in linea con le best practice per proteggere i dati nei bucket del tuo account o della tua organizzazione. Ad esempio, è possibile utilizzare i parametri per la protezione dei dati per identificare i bucket che non utilizzano chiavi AWS Key Management Service (AWS KMS) (SSE-KMS) per la crittografia predefinita o le richieste che utilizzano AWS Signature Version 2 (SigV2).

I seguenti casi d'uso forniscono strategie per l'utilizzo del pannello di controllo di S3 Storage Lens al fine di identificare i valori anomali e applicare le best practice per la protezione dei dati tra i bucket S3.

Argomenti

- Identificare i bucket che non utilizzano la crittografia lato server con AWS KMS per la crittografia predefinita (SSE-KMS)
- Identificare i bucket con il controllo delle versioni S3 abilitato
- Identificare le richieste che usano AWS Signature Version 2 (SigV2)
- Conteggiare il numero totale di regole di replica per ogni bucket
- Identificare la percentuale di byte con blocco degli oggetti

Identificare i bucket che non utilizzano la crittografia lato server con AWS KMS per la crittografia predefinita (SSE-KMS)

La crittografia predefinita di Amazon S3 consente di impostare il comportamento di crittografia predefinito di un bucket S3. Per ulteriori informazioni, consulta the section called “Impostazione della crittografia predefinita del bucket”.

È possibile utilizzare i parametri relativi al numero di bucket abilitati per SSE-KMS e alla percentuale di bucket abilitati per SSE-KMS per identificare i bucket che utilizzano la crittografia lato server con chiavi AWS KMS (SSE-KMS) come crittografia predefinita. S3 Storage Lens fornisce anche parametri per byte non crittografati, oggetti non crittografati, byte crittografati e oggetti crittografati. Per un elenco completo di parametri, consulta Glossario dei parametri di Amazon S3 Storage Lens.

È possibile analizzare i parametri di crittografia SSE-KMS nel contesto dei parametri di crittografia generali per identificare i bucket che non utilizzano SSE-KMS. Se desideri utilizzare SSE-KMS per tutti i bucket del tuo account o della tua organizzazione, puoi quindi aggiornare le impostazioni di crittografia predefinite per questi bucket in modo che utilizzino SSE-KMS. Oltre a SSE-KMS, puoi usare la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) o chiavi fornite dal cliente (SSE-C). Per ulteriori informazioni, consulta Protezione dei dati con la crittografia.

Fase 1: identificare i bucket che usano SSE-KMS per la crittografia predefinita

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Dashboards (Pannelli di controllo), scegli il pannello di controllo che desideri visualizzare.
4. Nella sezione Trends and distributions (Tendenze e distribuzioni), scegli % SSE-KMS enabled bucket count (% conteggio bucket abilitati per SSE-KMS) per il parametro principale e % encrypted bytes (% byte crittografati) per il parametro secondario.

   Il grafico Trend for date (Tendenza per [data]) viene aggiornato per visualizzare le tendenze per SSE-KMS e byte crittografati.
5. Per visualizzare informazioni più granulari a livello di bucket per SSE-KMS:

   a. Scegli un punto sul grafico. Apparirà un riquadro con le scelte per visualizzare informazioni dettagliate più granulari.

   b. Scegli la dimensione Buckets (Bucket). Quindi, scegliere Apply (Applica).

6. Nel grafico Distribution by buckets for date (Distribuzione per bucket per [data]), scegli il parametro SSE-KMS enabled bucket count (Conteggio bucket abilitati per SSE-KMS).
7. Ora puoi vedere quali bucket hanno SSE-KMS abilitato e quali no.

Fase 2: aggiornare le impostazioni predefinite di crittografia dei bucket

Dopo aver determinato quali bucket utilizzano SSE-KMS nel contesto del parametro % encrypted bytes (% byte crittografati), puoi identificare i bucket che non utilizzano SSE-KMS. Facoltativamente, puoi quindi accedere a questi bucket mediante la console S3 e aggiornare le loro impostazioni di crittografia predefinite affinché utilizzino SSE-KMS o SSE-S3. Per ulteriori informazioni, consulta Configurazione della crittografia predefinita.

Identificare i bucket con il controllo delle versioni S3 abilitato

Se attivata, la funzionalità Controllo versioni S3 conserva più versioni dello stesso oggetto che possono essere utilizzate per recuperare rapidamente i dati nel caso in cui un oggetto venga eliminato o sovrascritto accidentalmente. Puoi utilizzare il parametro Versioning-enabled bucket count (Conteggio bucket con controllo versioni abilitato) per vedere quali bucket utilizzano la funzionalità S3 di controllo delle versioni. Quindi, puoi usare la console S3 per abilitare la funzionalità S3 di controllo delle versioni per altri bucket.

Fase 1: identificare i bucket con il controllo delle versioni S3 abilitato

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione, scegli Storage Lens, Pannelli di controllo.
3. Nell'elenco Dashboards (Pannelli di controllo), scegli il pannello di controllo che desideri visualizzare.
4. Nella sezione Trends and distributions (Tendenze e distribuzioni), scegli Versioning-enabled bucket count (Conteggio bucket con controllo versioni abilitato) per il parametro principale e Buckets (Bucket) per il parametro secondario.

   Il grafico Trend for date (Tendenza per [data]) viene aggiornato con le tendenze per i bucket con la funzionalità S3 di controllo delle versioni abilitata. Subito sotto la riga delle tendenze, puoi vedere le sottosezioni Storage class distribution (Distribuzione classi di archiviazione) e Region distribution (Distribuzione regionale).

5. Per visualizzare informazioni dettagliate più granulari per tutti i bucket visualizzati nel grafico Trend for date (Tendenza per [data]) in modo da poter eseguire un'analisi più approfondita, procedi come segue:

   a. Scegli un punto sul grafico. Apparirà un riquadro con le scelte per visualizzare informazioni dettagliate più granulari.

   b. Scegli una dimensione da applicare ai tuoi dati per un'analisi più approfondita: Account, Regione AWS, Storage class (Classe di archiviazione) o Bucket. Quindi, scegliere Apply (Applica).

6. Nella sezione Bubble analysis by buckets for date (Analisi a bolle per bucket per [data]), scegli i parametri Versioning-enabled bucket count (Conteggio bucket con controllo versioni abilitato), Buckets (Bucket) e Active buckets (Bucket attivi).

   La sezione Bubble analysis by buckets for date (Analisi a bolle per bucket per [data]) viene aggiornata per visualizzare i dati relativi ai parametri selezionati. Puoi utilizzare questi dati per vedere quali bucket hanno la funzionalità S3 di controllo delle versioni abilitata nel contesto del numero totale di bucket. Nella sezione Bubble analysis by buckets for date (Analisi a bolle per bucket per [data]), è possibile tracciare i bucket su più dimensioni utilizzando tre parametri per rappresentare l'asse X, l'asse Y e la dimensione della bolla.

Fase 2: abilitare il controllo delle versioni S3

Dopo aver identificato i bucket in cui è abilitata la funzionalità S3 del controllo delle versioni, puoi identificare i bucket in cui il controllo delle versioni S3 non è mai stato abilitato o il cui controllo delle versioni è sospeso. Facoltativamente, puoi quindi abilitare il controllo delle versioni per questi bucket nella console S3. Per ulteriori informazioni, consulta Abilitazione della funzione Controllo delle versioni sui bucket.

Identificare le richieste che usano AWS Signature Version 2 (SigV2)

È possibile utilizzare la metrica All unsupported signature requests (Tutte le richieste di firma non supportate) per identificare le richieste che utilizzano AWS Signature Version 2 (SigV2). Questi dati possono aiutarti a identificare applicazioni specifiche che utilizzano SigV2. È possibile quindi eseguire la migrazione di queste applicazioni ad AWS Signature Version 4 (SigV4).

SigV4 è il metodo di firma consigliato per tutte le nuove applicazioni S3. SigV4 offre una maggiore sicurezza ed è supportato in tutte le Regioni AWS. Per ulteriori informazioni, consulta la sezione

relativa all'aggiornamento di Amazon S3 e all'estensione e alla modifica del periodo di obsolescenza di SigV2.

Prerequisito

Per visualizzare tutte le richieste di firma non supportate nel pannello di controllo di S3 Storage Lens, devi abilitare l'opzione S3 Storage Lens Advanced metrics and recommendations (Parametri e suggerimenti avanzati) e quindi selezionare Advanced data protection metrics (Parametri avanzati protezione dati). Per ulteriori informazioni, consulta Creazione e aggiornamento dei pannelli di controllo di Amazon S3 Storage Lens.

Fase 1: esaminare le tendenze di firma SigV2 per Account AWS, regione e bucket

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Dashboards (Pannelli di controllo), scegli il pannello di controllo che desideri visualizzare.
4. Per identificare bucket, account e regioni specifici con richieste che utilizzano SigV2:

    a. Nella sezione Top N overview for date (Panoramica primi N per [data]), in Top N (Primi N), inserisci il numero di bucket per i quali desideri visualizzare i dati.

    b. In Metric (Parametro), scegli All unsupported signature requests (Tutte le richieste di firma non supportate) nella categoria Data protection (Protezione dati).

    La sezione Top N overview for date (Panoramica primi N per [data]) viene aggiornata per visualizzare i dati per le richieste SigV2 per account, Regione AWS e bucket. La sezione Top N overview for date (Panoramica primi N per [data]) mostra anche la variazione percentuale rispetto al giorno o alla settimana precedente e un grafico sparkline per visualizzare la tendenza. Questa tendenza è valida per 14 giorni per i parametri gratuiti e per 30 giorni per i parametri e i suggerimenti avanzati.

    > **Note**
    >
    > Con i parametri avanzati e i suggerimenti di S3 Storage Lens, i parametri sono disponibili per le query per 15 mesi. Per ulteriori informazioni, consulta Selezione dei parametri.

Fase 2: identificare i bucket a cui le applicazioni accedono tramite richieste SigV2

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Dashboards (Pannelli di controllo), scegli il pannello di controllo che desideri visualizzare.
4. Nel pannello di controllo di Storage Lens, scegli la scheda Bucket.
5. Scorri fino alla sezione Buckets (Bucket). In Metrics categories (Categorie parametri), scegli Data protection (Protezione dati). Quindi deseleziona Summary (Riepilogo).

   L'elenco Buckets (Bucket) viene aggiornato per visualizzare tutti i parametri di protezione dei dati disponibili per i bucket visualizzati.
6. Per filtrare l'elenco Buckets (Bucket) in modo da visualizzare solo i parametri specifici di protezione dei dati, scegli l'icona delle preferenze ().
7. Deseleziona tutti i parametri di protezione dei dati finché non rimangono selezionati solo i seguenti parametri:

   - All unsupported signature requests (Tutte le richieste di firma non supportate)
   - % all unsupported signature requests (% tutte le richieste di firma non supportate)
8. (Facoltativo) In Page size (Dimensioni pagina), scegli il numero di bucket da visualizzare nell'elenco.
9. Scegli Confirm (Conferma).

   L'elenco Buckets (Bucket) viene aggiornato con i parametri a livello di bucket per le richieste SigV2. È possibile utilizzare questi dati per identificare bucket specifici con richieste SigV2. Quindi, puoi utilizzare queste informazioni per eseguire la migrazione delle tue applicazioni a SigV4. Per ulteriori informazioni, consulta la sezione Autenticazione delle richieste (AWS Signature Version 4) nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Conteggiare il numero totale di regole di replica per ogni bucket

La replica S3 consente di eseguire la copia asincrona e automatica di oggetti tra bucket Amazon S3. I bucket configurati per la replica di oggetti possono essere di proprietà dello stesso Account AWS o di account diversi. Per ulteriori informazioni, consulta Panoramica sulla replica degli oggetti.

Puoi utilizzare i parametri relativi al conteggio delle regole di replica di S3 Storage Lens per ottenere informazioni dettagliate per bucket sui bucket configurati per la replica. Queste informazioni includono le regole di replica all'interno di e tra bucket e regioni.

Prerequisito

Per visualizzare i parametri relativi al conteggio delle regole di replica nel pannello di controllo di S3 Storage Lens, devi abilitare l'opzione S3 Storage Lens Advanced metrics and recommendations (Parametri e suggerimenti avanzati) e quindi selezionare Advanced data protection metrics (Parametri avanzati protezione dati). Per ulteriori informazioni, consulta Creazione e aggiornamento dei pannelli di controllo di Amazon S3 Storage Lens.

Fase 1: contare il numero totale di regole di replica per ogni bucket

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Dashboards (Pannelli di controllo), scegli il pannello di controllo che desideri visualizzare.
4. Nel pannello di controllo di Storage Lens, scegli la scheda Bucket.
5. Scorri fino alla sezione Buckets (Bucket). In Metrics categories (Categorie parametri), scegli Data protection (Protezione dati). Quindi deseleziona Summary (Riepilogo).
6. Per filtrare l'elenco Buckets (Bucket) in modo da visualizzare solo i parametri relativi al conteggio delle regole di replica, scegli l'icona delle preferenze ().
7. Deseleziona tutti i parametri di protezione dei dati finché non rimangono selezionati solo i parametri relativi al conteggio delle regole di replica:

   - Same-Region Replication rule count (Conteggio regole di replica stessa regione)
   - Cross-Region Replication rule count (Conteggio regole di replica tra regioni)
   - Same-account replication rule count (Conteggio regole di replica stesso account)

- Cross-account replication rule count (Conteggio regole di replica tra account)
- Total replication rule count (Conteggio totale regole di replica)

8. (Facoltativo) In Page size (Dimensioni pagina), scegli il numero di bucket da visualizzare nell'elenco.
9. Scegli Confirm (Conferma).

Fase 2: aggiungere regole di replica

Dopo aver creato il conteggio delle regole di replica per bucket, facoltativamente è possibile creare altre regole di replica. Per ulteriori informazioni, consulta Esempi di configurazione della replica in tempo reale.

Identificare la percentuale di byte con blocco degli oggetti

La funzionalità S3 di blocco degli oggetti consente di archiviare gli oggetti utilizzando il modello write-once-read-many (WORM). Puoi usare il blocco degli oggetti per impedire che gli oggetti vengano eliminati o sovrascritti per un periodo di tempo fisso o indefinito. Puoi abilitare il blocco degli oggetti solo quando crei un bucket e abiliti anche la funzionalità S3 di controllo delle versioni. Tuttavia, puoi modificare il periodo di conservazione per le versioni dei singoli oggetti o applicare il blocco a fini legali per i bucket in cui è abilitato il blocco degli oggetti. Per ulteriori informazioni, consulta Utilizzo del blocco oggetti S3.

Puoi utilizzare i parametri di blocco degli oggetti in S3 Storage Lens per visualizzare il parametro % Object Lock in bytes (% blocco oggetti in byte) per il tuo account o la tua organizzazione. Puoi utilizzare queste informazioni per identificare i bucket non conformi alle best practice di protezione dei dati nel tuo account o nella tua organizzazione.

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Dashboards (Pannelli di controllo), scegli il pannello di controllo che desideri visualizzare.
4. Nella sezione Snapshot, in Metrics categories (Categorie parametri), scegli Data protection (Protezione dati).

La sezione Snapshot viene aggiornata per visualizzare i parametri di protezione dei dati, incluso il parametro % Object Lock in bytes (% blocco oggetti in byte). Puoi vedere la percentuale complessiva di byte con blocco degli oggetti per il tuo account o la tua organizzazione.

5. Per visualizzare il valore del parametro % Object Lock bytes (% blocco oggetti in byte) per bucket, scorri verso il basso fino alla sezione Top N overview (Panoramica primi N).

   Per ottenere dati a livello di oggetto per il blocco degli oggetti, puoi anche utilizzare i parametri Object Lock object count (Conteggio oggetti con blocco oggetti) e % Object Lock objects (% oggetti con blocco oggetti).

6. In Metric (Parametro), scegli % Object Lock bytes (% blocco oggetti in byte) nella categoria Data protection (Protezione dati).

   Per impostazione predefinita, la sezione Top N overview for date (Panoramica primi N per [data]) mostra i parametri per i primi 3 bucket. Nel campo Top N (Primi N) è possibile aumentare il numero di bucket. La sezione Top N overview for date (Panoramica primi N per [data]) mostra anche la variazione percentuale rispetto al giorno o alla settimana precedente e un grafico sparkline per visualizzare la tendenza. Questa tendenza è valida per 14 giorni per i parametri gratuiti e per 30 giorni per i parametri e i suggerimenti avanzati.

   > **Note**
   >
   > Con i parametri avanzati e i suggerimenti di S3 Storage Lens, i parametri sono disponibili per le query per 15 mesi. Per ulteriori informazioni, consulta Selezione dei parametri.

7. Controlla i seguenti dati per il parametro % Object Lock bytes (% blocco oggetti in byte):

   - Top number accounts (Primi [numero] account): verifica quali account hanno il valore più alto e il valore più basso per il parametro % Object Lock bytes (% blocco oggetti in byte).
   - Top number Regions (Prime [numero] regioni): visualizza un'analisi dettagliata dei valori del parametro % Object Lock bytes (% blocco oggetti in byte) per regione.
   - Top number buckets (Primi [numero] bucket): verifica quali bucket hanno il valore più alto e il valore più basso per il parametro % Object Lock bytes (% blocco oggetti in byte).

Utilizzo di S3 Storage Lens per controllare le impostazioni di Object Ownership

Amazon S3 Object Ownership è un'impostazione a livello di bucket S3 che puoi utilizzare per disabilitare le liste di controllo degli accessi (ACL) e controllare la proprietà degli oggetti nel bucket. Se la caratteristica Object Ownership è impostata su Bucket owner enforced (Applicata da proprietario bucket), puoi disabilitare le liste di controllo accessi (ACL) e assumere la proprietà di ogni oggetto incluso nel tuo bucket. Questo approccio semplifica la gestione degli accessi per i dati archiviati in Amazon S3.

Per impostazione predefinita, quando un altro Account AWS carica un oggetto nel bucket S3, tale account (l'object writer) possiede l'oggetto, ne ha accesso e può concedere ad altri utenti l'accesso tramite ACL. È possibile utilizzare Object Ownership per modificare questo comportamento di default.

La maggior parte degli attuali casi d'uso in Amazon S3 non richiede più l'uso delle ACL. Si consiglia pertanto di disabilitare le liste di controllo degli accessi, tranne nei casi in cui è necessario controllare individualmente l'accesso per ciascun oggetto. Se la caratteristica Object Ownership è impostata su Bucket owner enforced (Applicata da proprietario bucket), è possibile disabilitare le ACL e usare le policy per il controllo degli accessi. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

Con i parametri di gestione degli accessi di S3 Storage Lens, puoi identificare i bucket in cui le liste di controllo degli accessi non sono state disabilitate. Dopo aver identificato questi bucket, puoi eseguire la migrazione delle autorizzazioni ACL alle policy e disabilitare le ACL per questi bucket.

Argomenti

Fase 1: identificare le tendenze generali per le impostazioni di Object Ownership

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Dashboards (Pannelli di controllo), scegli il pannello di controllo che desideri visualizzare.

4. Nella sezione Snapshot for date (Snapshot per [data]), in Metrics categories (Categorie parametri), scegli Access management (Gestione accessi).

   La sezione Snapshot for date (Snapshot per [data]) viene aggiornata in modo da visualizzare il parametro % Object Ownership bucket owner enforced (% Object Ownership applicata da proprietario bucket). Puoi visualizzare la percentuale complessiva di bucket nell'account o organizzazione che usano l'impostazione Bucket owner enforced (Applicata da proprietario bucket) per Object Ownership per disabilitare le liste di controllo degli accessi.

Fase 2: identificare le tendenze a livello di bucket per le impostazioni di Object Ownership

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Dashboards (Pannelli di controllo), scegli il pannello di controllo che desideri visualizzare.
4. Per visualizzare parametri più dettagliati a livello di bucket, scegli la scheda Bucket.
5. Nella sezione Distribution by buckets for date (Distribuzione per bucket per [data]), scegli il parametro % Object Ownership bucket owner enforced (% Object Ownership applicata da proprietario bucket).

   Il grafico viene aggiornato per mostrare una ripartizione per bucket per il parametro % Object Ownership bucket owner enforced (% Object Ownership applicata da proprietario bucket). Puoi vedere quali bucket utilizzano l'impostazione Bucket owner enforced (Applicata da proprietario bucket) per la caratteristica Object Ownership per disabilitare le ACL.
6. Per visualizzare le impostazioni per Bucket owner enforced (Applicata da proprietario bucket) nel contesto, scorri verso il basso fino alla sezione Buckets (Bucket). In Metrics categories (Categorie parametri), seleziona Access management (Gestione accessi). Quindi deseleziona Summary (Riepilogo).

   Nell'elenco Buckets (Bucket) sono visualizzati i dati di tutte e tre le impostazioni di Object Ownership: Bucket owner enforced (Proprietario del bucket imposto), Bucket Owner Preferred (Proprietario preferito del bucket) e Object Writer.
7. Per filtrare l'elenco Buckets (Bucket) in modo da visualizzare i parametri metriche solo per una specifica impostazione di Object Ownership, scegli l'icona delle preferenze ().

8. Cancella i parametri che non desideri visualizzare.
9. (Facoltativo) In Page size (Dimensioni pagina), scegli il numero di bucket da visualizzare nell'elenco.
10. Scegli Confirm (Conferma).

Fase 3: aggiornare l'impostazione di Object Ownership su Bucket owner enforced (Applicata da proprietario bucket) per disabilitare le ACL

Dopo aver identificato i bucket che utilizzano l'impostazione Object Writer e Bucket Owner Preferred (Preferita da proprietario bucket) per Object Ownership, puoi eseguire la migrazione delle autorizzazioni ACL alle policy di bucket. Dopo aver completato la migrazione delle autorizzazioni ACL, puoi aggiornare l'impostazione di Object Ownership su Bucket owner enforced (Applicata da proprietario bucket) per disabilitare le ACL. Per ulteriori informazioni, consulta Prerequisiti per la disabilitazione delle ACL.

Utilizzo dei parametri di S3 Storage Lens per migliorare le prestazioni

Se hai abilitato l'opzione Advanced metrics (Parametri avanzati) di S3 Storage Lens, puoi utilizzare i parametri dei codici di stato dettagliati per ottenere i numeri delle richieste riuscite o non riuscite. È possibile utilizzare queste informazioni per risolvere i problemi relativi ad accesso e prestazioni. I parametri dei codici di stato dettagliati mostrano i conteggi dei codici di stato HTTP, come 403 Forbidden (403 Accesso negato) e 503 Service Unavailable (503 Servizio non disponibile). Puoi esaminare le tendenze generali relative ai parametri dei codici di stato dettagliati a livello di bucket S3, account e organizzazioni. Puoi quindi eseguire il drill-down dei parametri a livello di bucket per identificare i carichi di lavoro che attualmente accedono a questi bucket e causano errori.

Ad esempio, puoi esaminare il parametro 403 Forbidden error count (Conteggio errori 403 Accesso negato) per identificare i carichi di lavoro che accedono ai bucket senza le autorizzazioni corrette applicate. Dopo aver identificato questi carichi di lavoro, puoi eseguire un'analisi approfondita all'esterno di S3 Storage Lens per risolvere gli errori 403 Forbidden (403 Accesso negato).

Questo esempio mostra come eseguire un'analisi delle tendenze per l'errore 403 Forbidden (403 Accesso negato) utilizzando i parametri 403 Forbidden error count (Conteggio errori 403 Accesso negato) e % 403 Forbidden errors (% errori 403 Accesso negato). Puoi utilizzare questi parametri per identificare i carichi di lavoro che accedono ai bucket senza le autorizzazioni corrette applicate. Puoi eseguire un'analisi delle tendenze simile per qualsiasi altro parametro dei codici di stato dettagliati. Per ulteriori informazioni, consulta Glossario dei parametri di Amazon S3 Storage Lens.

Prerequisito

Per visualizzare il parametro Detailed status code metrics (Parametri codice di stato dettagliato) nel pannello di controllo di S3 Storage Lens, devi abilitare l'opzione S3 Storage Lens Advanced metrics and recommendations (Parametri avanzati e suggerimenti) e quindi selezionare Detailed status code metrics (Parametri codice di stato dettagliato). Per ulteriori informazioni, consulta Creazione e aggiornamento dei pannelli di controllo di Amazon S3 Storage Lens.

Argomenti

Fase 1: eseguire un'analisi delle tendenze per un singolo codice di stato HTTP

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Dashboards (Pannelli di controllo), scegli il pannello di controllo che desideri visualizzare.
4. Nella sezione Trends and distributions (Tendenze e distribuzioni), in Primary metric (Parametro principale), scegli 403 Forbidden error count (Conteggio errori 403 Accesso negato) nella categoria Detailed status codes (Codici di stato dettagliati). In Secondary metric (Parametro secondario), scegli % 403 Forbidden errors (% errori 403 Accesso negato).
5. Scorri verso il basso fino alla sezione Top N overview for date (Panoramica primi N per [data]). In Metrics (Parametri), scegli 403 Forbidden error count (Conteggio errori 403 Accesso negato) o % 403 Forbidden errors (% errori 403 Accesso negato) nella categoria Detailed status codes (Codici di stato dettagliati).

   La sezione Top N overview for date (Panoramica primi N per [data]) viene aggiornata per visualizzare i primi conteggi degli errori 403 Accesso negato per account, Regione AWS e bucket.

Fase 2: analizzare il conteggio degli errori per bucket

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Dashboards (Pannelli di controllo), scegli il pannello di controllo che desideri visualizzare.
4. Nel pannello di controllo di Storage Lens, scegli la scheda Bucket.
5. Scorri fino alla sezione Buckets (Bucket). In Metrics categories (Categorie parametri), seleziona il parametro Detailed status code (Codice di stato dettagliato). Quindi deseleziona Summary (Riepilogo).

   L'elenco Buckets (Bucket) viene aggiornato per visualizzare tutti i parametri dei codici di stato dettagliati disponibili. È possibile utilizzare queste informazioni per individuare i bucket con una elevata percentuale di codici di stato HTTP specifici e i codici di stato comuni tra bucket.
6. Per filtrare l'elenco Buckets (Bucket) in modo da visualizzare solo i parametri relativi a codici di stato dettagliati specifici, scegli l'icona delle preferenze ().
7. Deseleziona i parametri dei codici di stato dettagliati che non desideri visualizzare nell'elenco Buckets (Bucket).
8. (Facoltativo) In Page size (Dimensioni pagina), scegli il numero di bucket da visualizzare nell'elenco.
9. Scegli Confirm (Conferma).

   L'elenco Buckets (Bucket) mostra i parametri relativi al conteggio degli errori per il numero di bucket specificato. È possibile utilizzare queste informazioni per identificare bucket specifici che presentano molti errori e per risolvere gli errori per bucket.

Fase 3: correggere gli errori

Dopo aver identificato i bucket con una percentuale elevata di codici di stato HTTP specifici, è possibile risolvere questi errori. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Perché ricevo un errore 403 Forbidden (403 Accesso negato) quando tento di caricare file in Amazon S3?

- Perché ricevo un errore 403 Forbidden (403 Accesso negato) quando tento di modificare una policy di bucket in Amazon S3?
- Come posso correggere gli errori 403 Forbidden (403 Accesso negato) nel mio bucket Amazon S3 in cui tutte le risorse provengono dallo stesso Account AWS?
- Come posso risolvere un errore HTTP 500 o 503 in Amazon S3?

Glossario dei parametri di Amazon S3 Storage Lens

Il glossario dei parametri di Amazon S3 Storage Lens fornisce un elenco completo di parametri gratuiti e avanzati per S3 Storage Lens.

S3 Storage Lens offre parametri gratuiti per tutti i pannelli di controllo e le configurazioni con la possibilità di eseguire l'aggiornamento ai parametri avanzati.

- I parametri gratuiti contengono dati rilevanti per l'utilizzo dell'archiviazione, come il numero di bucket e gli oggetti nel tuo account. I parametri gratuiti includono anche parametri basati sui casi d'uso, come quelli relativi all'ottimizzazione dei costi e alla protezione dei dati. Tutti i parametri gratuiti vengono raccolti quotidianamente e i dati sono disponibili per le query per un massimo di 14 giorni.
- I parametri avanzati e i suggerimenti includono tutti i parametri gratuiti e i parametri aggiuntivi, ad esempio i parametri avanzati relativi alla protezione dei dati e all'ottimizzazione dei costi. I parametri avanzati includono anche categorie di parametri aggiuntive, come i parametri di attività e i parametri dettagliati relativi al codice di stato. I dati dei parametri avanzati sono disponibili per le query per 15 mesi.

  Per l'uso di S3 Storage Lens con le raccomandazioni e i parametri avanzati sono previsti costi aggiuntivi. Per ulteriori informazioni, consulta Prezzi di Amazon S3. Per ulteriori informazioni sui parametri avanzati e sulle funzioni di suggerimento, consulta Selezione dei parametri.

> **Note**
>
> Per i gruppi Storage Lens sono disponibili solo i parametri di archiviazione del piano gratuito. I parametri di livello avanzato non sono disponibili a livello di gruppo Storage Lens.

Nomi dei parametri

Nella colonna Nome parametro nella tabella seguente è riportato il nome di ogni parametro S3 Storage Lens nella console S3. Nella colonna CloudWatch ed esportazione è riportato il nome di ogni parametro in Amazon CloudWatch assieme al file di esportazione dei parametri che puoi configurare nel pannello di controllo di S3 Storage Lens.

Formule dei parametri derivati

I parametri derivati non sono disponibili per l'esportazione dei parametri e per l'opzione di pubblicazione CloudWatch. Puoi tuttavia usare le formula dei parametri riportate nella colonna Formule dei parametri derivati per calcolarli.

Interpretazione dei simboli dei prefissi per multipli di unità delle metriche di Amazon S3 Storage Lens (K, M, G e così via)

I multipli di unità di parametri di S3 Storage Lens sono scritti con simboli di prefisso. Questi simboli di prefisso sono rappresentati tramite i simboli del Sistema di unità internazionale (SI) standardizzati dall'International Bureau of Weights and Measures (BIPM). Vengono inoltre utilizzati nel codice unificato per le unità di misura (UCUM). Per ulteriori informazioni, consulta Elenco dei simboli dei prefissi SI.

> **Note**
>
> - L'unità di misura per i byte di archiviazione S3 è espressa in gigabyte binari (GB), dove 1 GB è pari a 2^{30} byte, 1 TB a 2^{40} byte e 1 PB a 2^{50} byte. Questa unità di misura è nota anche come gibibyte (GiB), come definito dalla Commissione elettrotecnica internazionale (IEC).
> - Quando un oggetto raggiunge la fine del suo ciclo di vita in base alla relativa configurazione, Amazon S3 lo aggiunge alla coda degli oggetti da eliminare e lo rimuove in modo asincrono. Deve pertanto esistere un ritardo tra la data di scadenza dell'oggetto e la data in cui Amazon S3 rimuove tale oggetto. S3 Storage Lens non include i parametri per gli oggetti scaduti e non ancora rimossi. Per ulteriori informazioni sulle operazioni di scadenza nel ciclo di vita S3, consulta Oggetti in scadenza.

Glossario dei parametri di S3 Storage Lens

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Archiviazione totale	StorageBytes	Lo spazio di archiviazione totale, inclusi i caricamenti incompleti in più parti, i metadati degli oggetti e i contrassegni di eliminazione	Gratu	Riep	N	-
Object count (Conteggio oggetti)	ObjectCount	Il numero totale degli oggetti	Gratu	Riep	N	-
Dimensione media degli oggetti	-	La dimension e media degli oggetti	Gratu	Riep	Y	sum(Stora geBytes)/ sum(Objec tCount)
Active buckets (Bucket attivi)	-	Il numero totale di bucket in uso attivo con archiviazione > 0 byte	Gratu	Riep	Y	-
Bucket	-	Numero totale di bucket	Gratu	Riep	Y	-
Account	-	Il numero di account il cui storage è nell'ambi to	Gratu	Riep	Y	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Current version bytes (Byte versione corrente)	CurrentVersionStorageBytes	Numero di byte che sono una versione corrente di un oggetto	Gratu	Ottin zion dei costi	N	-
% current version bytes (% byte versione corrente)	-	Percentuale di byte nell'ambito che sono versioni correnti degli oggetti	Gratu	Ottin zion dei costi	Y	sum(CurrentVersionStorageBytes)/sum(StorageBytes)
Current version object count (Conteggio oggetti versione corrente)	CurrentVersionObjectCount	Il numero degli oggetti della versione corrente	Gratu	Ottin zion dei costi	N	-
% current version objects (% oggetti versione corrente)	-	Percentuale di oggetti nell'ambito che sono una versione corrente	Gratu	Ottin zion dei costi	Y	sum(CurrentVersionObjectCount)/sum(ObjectCount)
Noncurrent version bytes (Byte versione non corrente)	NonCurrentVersionStorageBytes	Numero di byte versione non corrente	Gratu	Ottin zion dei costi	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
% noncurrent version bytes (% byte versione non corrente)	-	Percentuale di byte nell'ambito che sono versioni non correnti	Gratu	Ottin zion dei costi	Y	sum(NonCurrentVersionStorageBytes)/sum(StorageBytes)
Noncurrent version object count (Conteggio oggetti versione non corrente)	NonCurrentVersionObjectCount	Conteggio delle versioni non correnti dell'oggetto	Gratu	Ottin zion dei costi	N	-
% noncurrent version objects (% oggetti versione non corrente)	-	Percentuale di oggetti nell'ambito che sono una versione non corrente	Gratu	Ottin zion dei costi	Y	sum(NonCurrentVersionObjectCount)/sum(ObjectCount)
Delete marker bytes (Byte contrassegni di eliminazione)	DeleteMarkerStorageBytes	Numero di byte nell'ambito che sono contrassegni di eliminazione	Gratu	Ottin zion dei costi	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
% delete marker bytes (% byte contrassegni di eliminazione)	-	Percentuale di byte nell'ambito che sono contrassegni di eliminazione	Gratu	Ottin zion dei costi	Y	sum(DeleteMarkerStorageBytes)/sum(StorageBytes)
Delete marker object count (Conteggio oggetti contrassegni di eliminazione)	DeleteMarkerObjectCount	Il numero totale di oggetti con un contrassegno di eliminazione	Gratu	Ottin zion dei costi	N	-
% delete marker objects (% oggetti contrassegni di eliminazione)	-	La percentuale di oggetti nell'ambito con un contrassegno di eliminazione	Gratu	Ottin zion dei costi	Y	sum(DeleteMarkerObjectCount)/sum(ObjectCount)
Incomplete multipart upload bytes (Byte caricamenti in più parti incompleti)	IncompleteMultipartUploadStorageBytes	Byte totali nell'ambito per caricamenti in più parti incompleti	Gratu	Ottin zion dei costi	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
% incomplete multipart upload bytes (% byte caricamenti in più parti incompleti)	-	Percentuale di byte nell'ambito che sono il risultato di caricamenti in più parti incompleti	Gratu	Ottin zion dei costi	Y	sum(IncompleteMultipartUploadStorageBytes)/sum(StorageBytes)
Incomplete multipart upload object count (Conteggio oggetti caricamenti in più parti incompleti)	IncompleteMultipartUploadObjectCount	Il numero di oggetti nell'ambito che sono caricamenti in più parti incompleti	Gratu	Ottin zion dei costi	N	-
% incomplete multipart upload objects (% oggetti caricamenti in più parti incompleti)	-	La percentuale di oggetti nell'ambito che sono caricamenti in più parti incompleti	Gratu	Ottin zion dei costi	Y	sum(IncompleteMultipartUploadObjectCount)/sum(ObjectCount)

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Incomplete multipart upload storage bytes greater than 7 days old (Byte archiviazione caricamenti in più parti incompleti risalenti a più di 7 giorni)	IncompleteMPUStorageBytesOlderThan7Days	Byte totali relativi ai caricamenti in più parti incompleti che risalgono a più di 7 giorni	Gratu	Ottin zione dei costi	N	-
% incomplete multipart upload storage bytes greater than 7 days old (% byte archiviazione caricamenti in più parti incompleti risalenti a più di 7 giorni)	-	Percentuale di byte relativi ai caricamenti in più parti incompleti che risalgono a più di 7 giorni	Gratu	Ottin zione dei costi	Y	sum(IncompleteMPUStorageBytesOlderThan7Days)/sum(StorageBytes)

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Incomplete multipart upload object count greater than 7 days old (Conteggio oggetti caricamenti in più parti incompleti risalenti a più di 7 giorni)	IncompleteMPUObjectCountOlderThan7Days	Numero di oggetti che sono caricamenti in più parti incompleti che risalgono a più di 7 giorni	Gratu	Ottin zione dei costi	N	-
% incomplete multipart upload object count greater than 7 days old (% conteggio oggetti caricamenti in più parti incompleti risalenti a più di 7 giorni)	-	Percentuale di oggetti che sono caricamenti in più parti incompleti che risalgono a più di 7 giorni	Gratu	Ottin zione dei costi	Y	sum(IncompleteMPUObjectCountOlderThan7Days)/sum(ObjectCount)
Transition lifecycle rule count (Conteggio regole ciclo di vita transizione)	TransitionLifecycleRuleCount	Conteggio delle regole del ciclo di vita per la transizione degli oggetti a un'altra classe di archiviazione	Avan	Ottin zione dei costi	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Average transition lifecycle rules per bucket (Media regole ciclo di vita transizioni per bucket)	-	Numero medio di regole del ciclo di vita per la transizione degli oggetti a un'altra classe di archiviazione	Avan	Ottin zion dei costi	Y	sum(TransitionLifecycleRuleCount)/ sum(DistinctNumberOfBuckets)
Expiration lifecycle rule count (Conteggio regole ciclo di vita scadenza)	ExpirationLifecycleRuleCount	Conteggio delle regole del ciclo di vita che determinano la scadenza degli oggetti	Avan	Ottin zion dei costi	N	-
Average expiration lifecycle rules per bucket (Media regole ciclo di vita scadenza per bucket)	-	Numero medio di regole del ciclo di vita che determinano la scadenza degli oggetti	Avan	Ottin zion dei costi	Y	sum(ExpirationLifecycleRuleCount)/ sum(DistinctNumberOfBuckets)
Noncurrent version transition lifecycle rule count (Conteggio regole ciclo di vita transizione versioni non correnti)	NoncurrentVersionTransitionLifecycleRuleCount	Conteggio delle regole del ciclo di vita per la transizione delle versioni non correnti degli oggetti a un'altra classe di archiviazione	Avan	Ottin zion dei costi	N	

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Average noncurrent version transition lifecycle rules per bucket (Media regole ciclo di vita transizioni versioni non correnti per bucket)	-	Numero medio di regole del ciclo di vita per la transizione delle versioni non correnti degli oggetti a un'altra classe di archiviazione	Avan	Ottin zione dei costi	Y	sum(NoncurrentVersionTransitionLifecycleRuleCount)/sum(DistinctNumberOfBuckets)
Noncurrent version expiration lifecycle rule count (Conteggio regole ciclo di vita scadenza versioni non correnti)	NoncurrentVersionExpirationLifecycleRuleCount	Conteggio delle regole del ciclo di vita che fanno scadere le versioni non correnti degli oggetti	Avan	Ottin zione dei costi	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Average noncurrent version expiration lifecycle rules per bucket (Media regole ciclo di vita scadenza versioni non correnti per bucket)	-	Numero medio delle regole del ciclo di vita che fanno scadere le versioni non correnti degli oggetti	Avan	Ottin zion dei costi	Y	sum(NoncurrentVersionExpirationLifecycleRuleCount)/sum(DistinctNumberOfBuckets)
Abort incomplete multipart upload lifecycle rule count (Conteggio regole ciclo di vita interruzione caricamenti in più parti incompleti)	AbortIncompleteMPULifecycleRuleCount	Conteggio delle regole del ciclo di vita per eliminare i caricamenti in più parti incompleti	Avan	Ottin zion dei costi	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Average abort incomplete multipart upload lifecycle rules per bucket (Media interruzioni regole ciclo di vita caricamenti in più parti incompleti per bucket)	-	Numero medio delle regole del ciclo di vita per eliminare i caricamenti in più parti incompleti	Avan	Ottin zion dei costi	Y	sum(AbortIncompleteMPULifecycleRuleCount)/sum(DistinctNumberOfBuckets)
Expired object delete marker lifecycle rule count (Conteggio regole ciclo di vita contrassegni di eliminazione oggetti scaduti)	ExpiredObjectDeleteMarkerLifecycleRuleCount	Conteggio delle regole ciclo di vita per rimuovere i contrassegni di eliminazione degli oggetti scaduti	Avan	Ottin zion dei costi	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Average expired object delete marker lifecycle rules per bucket (Media regole ciclo di vita contrassegni di eliminazione oggetti scaduti per bucket)	-	Numero medio di regole ciclo di vita per rimuovere i contrassegni di eliminazione degli oggetti scaduti	Avan	Ottin zion dei costi	Y	sum(ExpiredObjectDeleteMarkerLifecycleRuleCount)/ sum(DistinctNumberOfBuckets)
Total lifecycle rule count (Conteggio totale regole ciclo di vita)	TotalLifecycleRuleCount	Conteggio totale delle regole del ciclo di vita	Avan	Ottin zion dei costi	N	-
Average lifecycle rule count per bucket (Media conteggio regole ciclo di vita per bucket)	-	Numero medio di regole del ciclo di vita	Avan	Ottin zion dei costi	Y	sum(TotalLifecycleRuleCount)/ sum(DistinctNumberOfBuckets)
Encrypted bytes (Byte crittografati)	EncryptedStorageBytes	Numero totale di byte crittografati	Grat	Prot e dei dati	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
% encrypted bytes (% byte crittografati)	-	Percentuale di byte totali che sono crittografati	Gratu	Prot e dei dati	Y	sum(EncryptedObjectCount)/sum(StorageBytes)
Encrypted object count (Conteggio oggetti crittografati)	EncryptedObjectCount	Conteggio totale degli oggetti crittografati	Gratu	Prot e dei dati	N	-
% encrypted objects (% oggetti crittografati)	-	Percentuale di oggetti crittografati	Gratu	Prot e dei dati	Y	sum(EncryptedStorageBytes)/sum(ObjectCount)
Unencrypted bytes (Byte non crittografati)	UnencryptedStorageBytes	Numero di byte non crittografati	Gratu	Prot e dei dati	Y	sum(StorageBytes) - sum(EncryptedStorageBytes)
% unencrypted bytes (% byte non crittografati)	-	Percentuale di byte non crittografati	Gratu	Prot e dei dati	Y	sum(UnencryptedStorageBytes)/sum(StorageBytes)

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Unencrypted object count (Conteggio oggetti non crittografati)	UnencryptedObjectCount	Conteggio totale degli oggetti non crittografati	Gratu	Prote dei dati	Y	sum(ObjectCount) - sum(EncryptedObjectCount)
% unencrypted objects (% oggetti non crittografati)	-	La percentuale di oggetti non crittografati	Gratu	Prote dei dati	Y	sum(UnencryptedStorageBytes)/ sum(ObjectCount)
Replicated storage bytes source (Origine byte di archiviazione replicati)	ReplicatedStorageBytesSource	Numero totale di byte replicati dal bucket di origine	Gratu	Prote dei dati	N	-
% replicated bytes source (% origine byte replicati)	-	Percentuale del totale di byte replicati dal bucket di origine	Gratu	Prote dei dati	Y	sum(ReplicatedStorageBytesSource)/ sum(StorageBytes)

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Replicated object count source (Origine conteggio oggetti replicati)	ReplicatedObjectCountSource	Conteggio degli oggetti replicati dal bucket di origine	Gratu	Prot e dei dati	N	-
% replicated objects source (% origine oggetti replicati)	-	Percentuale del totale di oggetti replicati dal bucket di origine	Gratu	Prot e dei dati	Y	sum(ReplicatedStorageObjectCount)/ sum(ObjectCount)
Replication storage bytes destination (Destinazione byte di archiviazione di replica)	ReplicatedStorageBytes	Numero totale di byte replicati nel bucket di destinazione	Gratu	Prot e dei dati	Y	-
% replicated bytes destination (% destinazione byte replicati)	-	Percentuale del totale di byte replicati nel bucket di destinazione	Gratu	Prot e dei dati	Y	sum(ReplicatedStorageBytes) / sum(StorageBytes)

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Replicated object count destination (Destinazione conteggio oggetti replicati)	ReplicatedObjectCount	Conteggio degli oggetti replicati nel bucket di destinazione	Gratu	Prote dei dati	Y	-
% replicated objects destination (% destinazione oggetti replicati)	-	Percentuale del totale di oggetti replicati nel bucket di destinazione	Gratu	Prote dei dati	Y	sum(ReplicatedObjectCount)/sum(ObjectCount)
Object Lock bytes (Byte blocco oggetti)	ObjectLockEnabledStorageBytes	Conteggio totale dei byte di archiviazione abilitati per il blocco degli oggetti	Gratu	Prote dei dati	Y	sum(UnencryptedStorageBytes)/sum(ObjectLockEnabledStorageCount) - sum(ObjectLockEnabledStorageBytes)

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
% Object Lock bytes (% byte blocco oggetti)	-	Percentuale di byte di archiviazione abilitati per il blocco degli oggetti	Gratu	Prote dei dati	Y	sum(ObjectLockEnabledStorageBytes)/sum(StorageBytes)
Object Lock object count (Conteggio oggetti con blocco oggetti)	ObjectLockEnabledObjectCount	Conteggio totale di oggetti con blocco oggetti	Gratu	Prote dei dati	Y	-
% Object Lock objects (% oggetti blocco oggetti)	-	Percentuale di oggetti totali con blocco oggetti abilitato	Gratu	Prote dei dati	Y	sum(ObjectLockEnabledObjectCount)/sum(ObjectCount)
Versioning-enabled bucket count (Conteggio bucket con controllo delle versioni abilitato)	VersioningEnabledBucketCount	Conteggio dei bucket con il controllo delle versioni S3 abilitato	Gratu	Prote dei dati	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
% versioning-enabled buckets (% bucket con controllo delle versioni abilitato)	-	Percentuale di bucket con il controllo delle versioni S3 abilitato	Gratu	Prot e dei dati	Y	sum(VersioningEnabledBucketCount)/ sum(DistinctNumberOfBuckets)
MFA delete-enabled bucket count (Conteggio bucket con eliminazione MFA abilitata)	MFADeleteEnabledBucketCount	Conteggio di bucket con l'eliminazione dell'autenticazione a più fattori (MFA) abilitata	Gratu	Prot e dei dati	N	-
% MFA delete-enabled bucket count (% conteggio bucket con eliminazione MFA abilitata)	-	Percentuale di bucket con l'eliminazione dell'autenticazione a più fattori (MFA) abilitata	Gratu	Prot e dei dati	Y	sum(MFADeleteEnabledBucketCount)/ sum (DistinctNumberOfBuckets)

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Conteggio bucket con SSE-KMS abilitata	SSEKMSEnabledBucketCount	Conteggio dei bucket che utilizzano la crittografia lato server (SSE) con chiavi AWS Key Management Service (SSE-KMS) per la crittografia dei bucket predefinita	Gratu	Prot e dei dati	N	-
% SSE-KMS enabled buckets (% bucket con SSE-KMS abilitata)	-	Percentuale di bucket com SSE-KMS per crittografia bucket predefinita	Gratu	Prot e dei dati	Y	sum(SSEKMSEnabledBucketCount)/ sum(DistinctNumberOfBuckets)
All unsupported signature requests (Tutte le richieste di firma non supportate)	AllUnsupportedSignatureRequests	Numero totale di richieste che utilizzano versioni di firma AWS non supportate	Avan	Prot e dei dati	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
% all unsupported signature requests (% tutte le richieste di firma non supportate)	-	Percentuale di richieste che utilizzano versioni di firma AWS non supportate	Avan	Prot e dei dati	Y	sum(AllUnsupportedSignatureRequests) / sum(AllRequests)
All unsupported TLS requests (Tutte le richieste TLS non supportate)	AllUnsupportedTLSRequests	Numero di richieste che utilizzano versioni di Transport Layer Security (TLS) non supportate	Avan	Prot e dei dati	N	-
% all unsupported TLS requests (% tutte le richieste TLS non supportate)	-	Percentuale di richieste che utilizzano versioni TLS non supportate	Avan	Prot e dei dati	Y	sum(AllUnsupportedTLSRequests)/ sum(AllRequests)
All SSE-KMS requests (Tutte le richieste SSE-KMS)	AllSSEKMSRequests	Numero totale di richieste che specificano SSE-KMS	Avan	Prot e dei dati	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
% all SSE-KMS requests (% tutte le richieste SSE-KMS)	-	Percentuale di richieste che specificano SSE-KMS	Avan	Prot e dei dati	Y	sum(AllSSEKMSRequests)/sum(AllRequests)
Same-Region Replication rule count (Conteggio regole di replica stessa regione)	SameRegionReplicationRuleCount	Conteggio delle regole di replica per la replica nella stessa regione (SRR)	Avan	Prot e dei dati	N	-
Average Same-Region Replication rules per bucket (Media regole di replica nella stessa regione per bucket)	-	Numero medio di regole di replica per SRR	Avan	Prot e dei dati	Y	sum(SameRegionReplicationRuleCount)/sum(DistinctNumberOfBuckets)
Cross-Region Replication rule count (Conteggio regole di replica tra regioni)	CrossRegionReplicationRuleCount	Conteggio delle regole di replica per la replica tra regioni (CRR)	Avan	Prot e dei dati	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Average Cross-Region Replication rules per bucket (Media regole di replica tra regioni per bucket)	-	Numero medio di regole di replica per CRR	Avan	Prot e dei dati	Y	sum(CrossRegionReplicationRuleCount) / sum(DistinctNumberOfBuckets)
Same-account replication rule count (Conteggio regole di replica stesso account)	SameAccountReplicationRuleCount	Conteggio delle regole di replica per la replica all'interno dello stesso account	Avan	Prot e dei dati	N	-
Average same-account replication rules per bucket (Media regole di replica stesso account per bucket)	-	Numero medio di regole di replica per la replica all'interno dello stesso account	Avan	Prot e dei dati	Y	sum(SameAccountReplicationRuleCount) / sum(DistinctNumberOfBuckets)
Cross-account replication rule count (Conteggio regole di replica tra account)	CrossAccountReplicationRuleCount	Conteggio delle regole di replica per la replica tra account	Avan	Prot e dei dati	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Average cross-account replication rules per bucket (Media regole di replica tra account per bucket)	-	Numero medio di regole di replica per la replica tra account	Avan	Prot e dei dati	Y	sum(CrossAccountReplicationRuleCount)/ sum(DistinctNumberOfBuckets)
Invalid destination replication rule count (Conteggio regole di replica di destinazione non valida)	InvalidDestinationReplicationRuleCount	Conteggio delle regole di replica con una destinazione di replica non valida	Avan	Prot e dei dati	N	-
Average invalid destination replication rules per bucket (Media regole di replica destinazione non valida per bucket)	-	Numero medio di regole di replica con una destinazione di replica non valida	Avan	Prot e dei dati	Y	sum(InvalidReplicationRuleCount)/ sum (DistinctNumberOfBuckets)
Total replication rule count (Conteggio totale regole di replica)	-	Conteggio totale delle regole di replica	Avan	Prot e dei dati	Y	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Average replication rule count per bucket (Media conteggio regole di replica per bucket)	-	Media del conteggio totale delle regole di replica	Avan	Prot e dei dati	Y	sum(all replicati on rule count metrics)/ sum(Disti nctNumber OfBuckets)
Object Ownership bucket owner enforced bucket count (Numero di bucket con Object Ownership impostata su Bucket owner enforced [Applicata da proprietario bucket])	ObjectOwn ershipBuc ketOwnerE nforcedBu cketCount	Il numero totale di bucket con le liste di controllo degli accessi (ACL) disabilitate mediante l'uso dell'impostazione Bucket owner enforced (Applicat a da proprietario bucket) per Object Ownership	Grat	Gest degl acce	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
% Object Ownership bucket owner enforced buckets (Bucket con % Object Ownership applicata da proprietario bucket)	-	La percentuale di bucket con ACL disabilitate mediante l'uso dell'impostazione Bucket owner enforced (Applicata da proprietario bucket) per Object Ownership.	Gratu	Gest degl acce	Y	sum(ObjectOwnershipBucketOwnerEnforcedBucketCount)/ sum (Distinct NumberOfBuckets)
Object Ownership bucket owner preferred bucket count (Numero di bucket con Object Ownership impostata su Bucket Owner Preferred [Preferita da proprietario bucket])	ObjectOwnershipBucketOwnerPreferredBucketCount	Il numero totale di bucket che utilizzano l'impostazione Bucket Owner Preferred (Preferita da proprietario bucket) per Object Ownership	Gratu	Gest degl acce	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
% Object Ownership bucket owner preferred buckets (Bucket con % Object Ownership preferita da proprietario bucket)	-	La percentuale di bucket che utilizzano l'impostazione Bucket Owner Preferred (Preferita da proprietario bucket) per Object Ownership	Gratu	Gest degl acce	Y	sum(ObjectOwnershipBucketOwnerPreferredBucketCount)/ sum(DistinctNumberOfBuckets)
Object Ownership object writer bucket count (Conteggio bucket object writer Object Ownership)	ObjectOwnershipObjectWriterBucketCount	Conteggio totale di bucket che utilizzano l'impostazione object writer per Object Ownership	Gratu	Gest degl acce	N	-
% Object Ownership object writer buckets (% bucket object writer Object Ownership)	-	Percentuale di bucket che utilizzano l'impostazione object writer per Object Ownership	Gratu	Gest degl acce	Y	sum(ObjectOwnershipObjectWriterBucketCount)/ sum(DistinctNumberOfBuckets)

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Transfer Acceleration enabled bucket count (Numero di bucket con Transfer Acceleration abilitata)	TransferAccelerationEnabledBucketCount	Conteggio totale di bucket con Transfer Acceleration abilitata	Gratu	Presni	N	-
% Transfer Acceleration enabled buckets (% bucket con Transfer Acceleration abilitata)	-	Percentuale di bucket con Transfer Acceleration abilitata	Gratu	Presni	Y	sum(TransferAccelerationEnabledBucketCount)/sum(DistinctNumberOfBuckets)
Event Notification enabled bucket count (Conteggio di bucket con alla notifica eventi abilitata)	EventNotificationEnabledBucketCount	Conteggio totale di bucket con notifiche eventi abilitate	Gratu	Ever	N	

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
% Event Notification enabled buckets (% bucket con notifica eventi abilitata)	-	Percentuale di bucket con notifiche eventi abilitate	Gratu	Ever	Y	sum(EventNotificationEnabledBucketCount)/sum(DistinctNumberOfBuckets)
Tutte le richieste	AllRequests	Numero totale di richieste effettuate	Avan	Attiv	N	-
Richieste GET	GetRequests	Numero totale di richieste `GET` effettuate	Avan	Attiv	N	-
Put requests (Richieste PUT)	PutRequests	Numero totale di richieste `PUT` effettuate	Avan	Attiv	N	-
Head requests (Richieste HEAD)	HeadRequests	Numero totale di richieste `HEAD` effettuate	Avan	Attiv	N	-
Delete requests (Richieste DELETE)	DeleteRequests	Numero totale di richieste `DELETE` effettuate	Avan	Attiv	N	-
Richieste LIST	ListRequests	Numero totale di richieste `LIST` effettuate	Avan	Attiv	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
Post requests (Richieste POST)	PostRequests	Numero totale di richieste POST effettuate	Avan	Attiv	N	-
Select requests (Richieste Select)	SelectRequests	Numero totale di richieste S3 Select	Avan	Attiv	N	-
Select scanned bytes (Byte Select scansionati)	SelectScannedBytes	Numero di byte S3 Select scansionati	Avan	Attiv	N	-
Select returned bytes (Byte Select restituiti)	SelectReturnedBytes	Numero di byte S3 Select restituiti	Avan	Attiv	N	-
Byte scaricati	BytesDownloaded	Numero di byte scaricati	Avan	Attiv	N	-
% retrieval rate (% tasso di recupero)	-	Percentuale di byte scaricati	Avan	Attiv	Y	sum(BytesDownloaded)/sum(StorageBytes)
Byte caricati	BytesUploaded	Il numero di byte caricati	Avan	Attiv	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
% ingest ratio (% rapporto di acquisizione)	-	Percentuale di byte caricati	Avan	Attiv	Y	sum(BytesUploaded) / sum(StorageBytes)
4xx errors (Errori 4xx)	4xxErrors	Numero totale di codici di stato HTTP 4xx	Avan	Attiv	N	-
5xx errors (Errori 5xx)	5xxErrors	Numero totale di codici di stato HTTP 5xx	Avan	Attiv	N	-
Total errors (Totale errori)	-	Somma di tutti gli errori 4xx e 5xx	Avan	Attiv	Y	sum(4xxErrors) + sum(5xxErrors)
% error rate (% tasso di errore)	-	Numero totale di errori 4xx e 5xx come percentuale del totale delle richieste	Avan	Attiv	Y	sum(TotalErrors)/ sum(TotalRequests)
200 OK status count (Conteggio dello stato 200 OK)	200OKStatusCount	Conteggio totale dei codici di stato 200 OK	Avan	Codi di statc detta to	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
% 200 OK status (% stato 200 OK)	-	Numero totale di codici di stato 200 OK come percentuale delle richieste totali	Avan	Codi di statc detta to	Y	sum(200OK StatusCou nt)/ sum(A llRequest s)
206 Partial Content status count (Conteggio stato 206 contenuto parziale)	206Partia lContentS tatusCount	Conteggio totale dei codici di stato 206 contenuto parziale	Avan	Codi di statc detta to	N	-
% 206 Partial Content status (% stato 206 contenuto parziale)	-	Numero totale di codici di stato 206 contenuto parziale come percentua le delle richieste totali	Avan	Codi di statc detta to	Y	sum(206Pa rtialCont entStatus Count)/ su m(AllRequ ests)
400 Bad Request error count (Conteggio errori 400 Richiesta non valida)	400BadReq uestErrorCount	Conteggio totale dei codici di stato 400 Richiesta non valida	Avan	Codi di statc detta to	N	-

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
% 400 Bad Request errors (% errori 400 Richiesta non valida)	-	Numero totale di codici di stato 400 Richiesta non valida come percentuale delle richieste totali	Avan	Codi di statc detta to	Y	sum(400BadRequestErrorCount)/ sum(AllRequests)
403 Forbidden error count (Conteggio errori 403 Accesso negato)	403ForbiddenErrorCount	Conteggio totale dei codici di stato 403 Accesso negato	Avan	Codi di statc detta to	N	-
% 403 Forbidden errors (% errori 403 Accesso negato)	-	Numero totale di codici di stato 403 Accesso negato come percentuale delle richieste totali	Avan	Codi di statc detta to	Y	sum(403ForbiddenErrorCount) / sum(AllRequests)
404 Not Found error count (Conteggio errori 404 Non trovato)	404NotFoundErrorCount	Conteggio totale dei codici di stato 404 Non trovato	Avan	Codi di statc detta to	N	-
% 404 Not Found errors (% errori 404 Non trovato)	-	Numero totale di codici di stato 404 Non trovato come percentuale delle richieste totali	Avan	Codi di statc detta to	Y	sum(404NotFoundErrorCount)/ sum(AllRequests)

Nome parametro	CloudWatch ed esportazione	Descrizione	Livel	Cate[2]	D	Form para deriv
500 Internal Server Error count (Conteggio errori 500 Errore interno del server)	500InternalServerErrorCount	Conteggio totale dei codici di stato 500 Errore interno del server	Avan	Codi di statc detta to	N	-
% 500 Internal Server Errors (% errori 500 Errore interno del server)	-	Numero totale di codici di stato 500 Errore interno del server come percentuale delle richieste totali	Avan	Codi di statc detta to	Y	sum(500InternalServerErrorCount)/ sum (AllRequests)
503 Service Unavailable error count (Conteggio errori 503 Servizio non disponibile)	503ServiceUnavailableErrorCount	Conteggio totale dei codici di stato 503 Servizio non disponibile	Avan	Codi di statc detta to	N	-
% 503 Service Unavailable errors (% errori 503 Servizio non disponibile)	-	Numero totale di codici di stato 503 Servizio non disponibile come percentuale delle richieste totali	Avan	Codi di statc detta to	Y	sum(503ServiceUnavailableErrorCount) / sum(AllRequests)

[1]Tutti i parametri di archiviazione del piano gratuito non sono disponibili a livello di gruppo Storage Lens. I parametri di livello avanzato non sono disponibili a livello di gruppo Storage Lens.

[2] I parametri relativi al conteggio delle regole e quelli relativi alle impostazioni dei bucket non sono disponibili a livello di prefisso.

Utilizzo di Amazon S3 Storage Lens con la console e l'API

Amazon S3 Storage Lens è una funzionalità di analisi dell'archiviazione su cloud che puoi utilizzare per avere una panoramica completa a livello di organizzazione sull'utilizzo e sulle attività relative all'archiviazione di oggetti. È possibile utilizzare i parametri di S3 Storage Lens per generare approfondimenti, ad esempio per scoprire la quantità di spazio di archiviazione disponibile nell'intera organizzazione o quali sono i bucket e i prefissi caratterizzati da una crescita più rapida. Puoi anche utilizzare i parametri di S3 Storage Lens per individuare le opportunità di ottimizzazione dei costi, implementare le best practice di protezione e sicurezza dei dati e migliorare le prestazioni dei carichi di lavoro delle applicazioni. Ad esempio, puoi identificare i bucket che non hanno regole del ciclo di vita S3 per far scadere i caricamenti in più parti incompleti che risalgono a più di 7 giorni. Puoi anche individuare i bucket non conformi alle best practice di protezione dei dati, come quelli che usano la replica S3 o il controllo delle versioni S3. S3 Storage Lens analizza i parametri di archiviazione per fornire raccomandazioni contestuali che puoi usare per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati.

S3 Storage Lens aggrega i tuoi parametri e mostra le informazioni nella sezione Account snapshot (Snapshot dell'account) nella pagina Buckets (Bucket) della console di Amazon S3. S3 Storage Lens dispone inoltre di un pannello di controllo interattivo che puoi usare per visualizzare informazioni dettagliate e tendenze, contrassegnare le anomalie e ricevere consigli per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati. Nel pannello di controllo sono disponibili opzioni di drill-down per generare e visualizzare approfondimenti a livello di organizzazione, account, Regione AWS, classe di storage, bucket, prefisso o gruppo Storage Lens. Puoi anche inviare un'esportazione dei parametri quotidiana in formato CSV o Parquet a un bucket S3.

Questa sezione contiene esempi di creazione, aggiornamento e visualizzazione delle configurazioni S3 Storage Lens e l'esecuzione di operazioni correlate alla funzione. Se si utilizza S3 Storage Lens con AWS Organizations, questi esempi coprono anche questi casi d'uso. Negli esempi, sostituisci i valori delle variabili con valori adatti alle proprie esigenze.

Argomenti

Utilizzo di Amazon S3 Storage Lens sulla console

Amazon S3 Storage Lens è una funzionalità di analisi dell'archiviazione su cloud che puoi utilizzare per avere una panoramica completa a livello di organizzazione sull'utilizzo e sulle attività relative all'archiviazione di oggetti. È possibile utilizzare i parametri di S3 Storage Lens per generare approfondimenti, ad esempio per scoprire la quantità di spazio di archiviazione disponibile nell'intera organizzazione o quali sono i bucket e i prefissi caratterizzati da una crescita più rapida. Puoi anche utilizzare i parametri di S3 Storage Lens per individuare le opportunità di ottimizzazione dei costi, implementare le best practice di protezione e sicurezza dei dati e migliorare le prestazioni dei carichi di lavoro delle applicazioni. Ad esempio, puoi identificare i bucket che non hanno regole del ciclo di vita S3 per far scadere i caricamenti in più parti incompleti che risalgono a più di 7 giorni. Puoi anche individuare i bucket non conformi alle best practice di protezione dei dati, come quelli che usano la replica S3 o il controllo delle versioni S3. S3 Storage Lens analizza i parametri di archiviazione per fornire raccomandazioni contestuali che puoi usare per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati.

S3 Storage Lens aggrega i tuoi parametri e mostra le informazioni nella sezione Account snapshot (Snapshot dell'account) nella pagina Buckets (Bucket) della console di Amazon S3. S3 Storage Lens dispone inoltre di un pannello di controllo interattivo che puoi usare per visualizzare informazioni dettagliate e tendenze, contrassegnare le anomalie e ricevere consigli per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati. Nel pannello di controllo sono disponibili opzioni di drill-down per generare e visualizzare approfondimenti a livello di organizzazione, account, Regione AWS, classe di archiviazione, bucket, prefisso o gruppo Storage Lens. Puoi anche inviare un'esportazione dei parametri quotidiana in formato CSV o Parquet a un bucket S3.

> **Note**
>
> Affinché la visualizzazione di qualsiasi aggiornamento alla configurazione del pannello di controllo sia accurata, possono essere necessarie fino a 48 ore.

Argomenti

Creazione e aggiornamento dei pannelli di controllo di Amazon S3 Storage Lens

S3 Storage Lens aggrega i tuoi parametri e mostra le informazioni nella sezione Account snapshot (Snapshot dell'account) nella pagina Buckets (Bucket) della console Amazon S3. S3 Storage Lens dispone inoltre di un pannello di controllo interattivo che puoi usare per visualizzare informazioni dettagliate e tendenze, contrassegnare le anomalie e ricevere consigli per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati. Nel pannello di controllo sono disponibili opzioni di drill-down per generare e visualizzare approfondimenti a livello di organizzazione, account, Regione AWS, classe di archiviazione, bucket, prefisso o gruppo Storage Lens. Puoi anche inviare un'esportazione dei parametri quotidiana in formato CSV o Parquet a un bucket S3.

La dashboard predefinita di Amazon S3 Storage Lens è. default-account-dashboard Questo pannello di controllo è preconfigurato da Amazon S3 per aiutarti a visualizzare informazioni dettagliate di riepilogo e tendenze per i parametri avanzati e gratuiti aggregati dell'intero account nella console. Non puoi modificare l'ambito di configurazione del pannello di controllo predefinito, ma puoi aggiornare la selezione dei parametri dai parametri gratuiti ai suggerimenti e ai parametri avanzati a pagamento, configurare l'esportazione facoltativa dei parametri o addirittura disabilitare il pannello di controllo predefinito. Il pannello di controllo predefinito non può essere eliminato.

Puoi anche creare dashboard personalizzati S3 Storage Lens aggiuntivi che possono essere adattati alla tua organizzazione all'interno o a regioni AWS Organizations o gruppi specifici all'interno di un account.

Creazione di un pannello di controllo di Amazon S3 Storage Lens

Attieniti alla procedura seguente per creare un pannello di controllo Amazon S3 Storage Lens sulla console Amazon S3.

Fase 1: definire l'ambito del pannello di controllo

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nella barra di navigazione nella parte superiore della pagina, scegli il nome della AWS regione attualmente visualizzata. Quindi, scegli la regione a cui vuoi passare.
3. Nel pannello di navigazione a sinistra, in S3 Storage Lens scegli Pannelli di controllo.
4. Seleziona Crea pannello di controllo.
5. Nella sezione Generale della pagina Pannello di controllo completa le seguenti operazioni:

a. Visualizza la Home Region per la tua dashboard. La Home Region è la regione Regione AWS in cui sono archiviate la configurazione e le metriche di questa dashboard di Storage Lens.

b. Specifica il nome di un pannello di controllo.

   I nomi del pannello di controllo devono contenere meno di 65 caratteri e non possono contenere caratteri speciali o spazi.

   > **Note**
   >
   > Il nome del pannello di controllo dopo la creazione non potrà più essere modificato.

c. Facoltativamente, puoi decidere di aggiungere tag al pannello di controllo. I tag possono essere utilizzati per gestire le autorizzazioni per il pannello di controllo e tenere traccia dei costi per S3 Storage Lens.

   Per ulteriori informazioni, consulta Controllo dell'accesso mediante i tag di risorse nella Guida per l'utente di IAM e Tag per l'allocazione dei costi generati da AWS nella Guida per l'utente di AWS Billing .

   > **Note**
   >
   > Puoi aggiungere fino a 50 tag alla configurazione del pannello di controllo.

6. Nella sezione Ambito del pannello di controllo completa le seguenti operazioni:

   a. Seleziona le regioni e i bucket che desideri siano incluse o escluse da S3 Storage Lens nel pannello di controllo.

   b. Scegli i bucket nelle regioni selezionate che desideri siano inclusi o esclusi da S3 Storage Lens. Puoi includere o escludere i bucket, ma non puoi eseguire entrambe le operazioni. Questa opzione non è disponibile quando si creano pannelli di controllo a livello di organizzazione.

> **Note**
>
> - Puoi anche includere o escludere regioni e e bucket. Questa opzione è limitata alle regioni solo se si creano pannelli di controllo a livello di organizzazione tra gli account membri dell'organizzazione.
> - Puoi scegliere fino a 50 bucket da includere o escludere.

Fase 2: configurare la selezione dei parametri

1. Nella sezione Selezione parametri seleziona il tipo parametri che desideri aggregare per il pannello di controllo.

   - Per includere i parametri gratuiti aggregati a livello di bucket e disponibili per le query per 14 giorni, scegli Free metrics (Parametri gratuiti).
   - Per abilitare i parametri avanzati e altre opzioni avanzate, scegli Advanced metrics and recommendations (Parametri e suggerimenti avanzati). Queste opzioni includono l'aggregazione avanzata dei prefissi, la CloudWatch pubblicazione su Amazon e i consigli contestuali. I dati sono disponibili per le query per 15 mesi. I parametri e i suggerimenti avanzati hanno un costo aggiuntivo. Per ulteriori informazioni, consulta Prezzi di Amazon S3.

     Per ulteriori informazioni su parametri avanzati e parametri gratuiti, consulta Selezione dei parametri.
2. In Advanced metrics and recommendations features (Parametri avanzati e funzioni di suggerimento), seleziona le opzioni da abilitare:

   - Advanced metrics (Parametri avanzati)
   - CloudWatch pubblicazione
   - Aggregazione di prefisso

> **Important**
>
> Se abiliti l'aggregazione dei prefissi per la configurazione di S3 Storage Lens, le metriche a livello di prefisso non verranno pubblicate su. CloudWatch Vengono pubblicate solo le metriche di S3 Storage Lens a livello di bucket, account e organizzazione. CloudWatch

3. Se hai abilitato Advanced metrics (Parametri avanzati), in Advanced metrics categories (Categorie parametri avanzati) seleziona le categorie che desideri visualizzare nel pannello di controllo di S3 Storage Lens:

   - Parametri delle attività
   - Detailed status code metrics (Parametri dettagliati codice di stato)
   - Advanced cost optimization metrics (Parametri avanzati ottimizzazione costi)
   - Advanced data protection metrics (Parametri avanzati protezione dati)

   Per ulteriori informazioni sulle categorie di parametri, consulta Categorie di parametri. Per un elenco completo di parametri, consulta Glossario dei parametri di Amazon S3 Storage Lens.

4. Se hai scelto di abilitare l'aggregazione dei prefissi, configura quanto segue:

   a. Scegli la dimensione minima della soglia del prefisso per questo pannello di controllo.

      Ad esempio, una soglia di prefisso del 5% indica che verranno aggregati i prefissi che costituiscono una dimensione pari o superiore al 5% dell'archiviazione del bucket.

   b. Dovrai scegliere anche la profondità del prefisso.

      Questa impostazione indica il numero massimo di livelli fino a cui vengono valutati i prefissi. La profondità del prefisso deve essere inferiore a 10.

   c. Specifica un carattere delimitatore per il prefisso.

      Questo valore viene utilizzato per identificare ogni livello di prefisso. Il valore predefinito in Amazon S3 è il carattere `/`, ma la struttura dell'archiviazione potrebbe utilizzare altri caratteri delimitatori.

(Facoltativo) Fase 3: esportare i parametri per il pannello di controllo

1. Nella sezione Metrics export (Esportazione parametri), per creare un'esportazione dei parametri che verrà inserita quotidianamente in un bucket di destinazione a tua scelta scegli Enable (Abilita).

   I parametri vengono esportati in formato CSV o Apache Parquet. Essa rappresenta lo stesso ambito di dati dei dati del pannello di controllo di S3 Storage Lens senza le raccomandazioni.
2. Se hai abilitato l'esportazione dei parametri, scegli il relativo formato di output, ovvero CSV o Apache Parquet.

   Parquet è un formato di file open source per Hadoop che memorizza i dati nidificati in un formato a colonne semplice.
3. Scegli il bucket S3 di destinazione per l'esportazione dei parametri.

   Puoi scegliere un bucket nell'account corrente del pannello di controllo di S3 Storage Lens. Oppure puoi sceglierne un altro Account AWS se disponi delle autorizzazioni per il bucket di destinazione e dell'ID account del proprietario del bucket di destinazione.
4. Scegli il bucket S3 di destinazione (formato: `s3://`*`bucket-name`*`/`*`prefix`*).

   Il bucket deve trovarsi nella regione principale del pannello di controllo di S3 Storage Lens. Nella casella Destination bucket permission (Autorizzazione bucket di destinazione) della console S3 verrà visualizzata l'autorizzazione che verrà aggiunta da Amazon S3 alla policy di bucket di destinazione. Amazon S3 aggiornerà la policy relative ai bucket sul bucket di destinazione per consentire a S3 di inserire i dati in quel bucket.
5. (Facoltativo) Per abilitare la crittografia lato server per l'esportazione dei parametri, segli Specify an encryption key (Specifica una chiave di crittografia). Quindi scegli il Tipo di crittografia: Chiavi gestite da Amazon S3 (SSE-S3) o Chiave AWS Key Management Service (SSE-KMS).

   Puoi scegliere una chiave gestita da Amazon S3 (SSE-S3) o una chiave AWS Key Management Service (AWS KMS) (SSE-KMS).
6. (Facoltativo) Per specificare una AWS KMS chiave, devi scegliere una chiave KMS o inserire una chiave Amazon Resource Name (ARN).

   Se scegli una chiave gestita dal cliente, devi concedere a S3 Storage Lens l'autorizzazione a eseguire la crittografia nella policy della chiave AWS KMS . Per ulteriori informazioni, consulta Utilizzo di un file AWS KMS key per crittografare le esportazioni delle metriche.
7. Seleziona Crea pannello di controllo.

Per ottenere ulteriore visibilità sull'archiviazione, è possibile creare uno o più gruppi S3 Storage Lens e collegarli al pannello di controllo. Un gruppo S3 Storage Lens è un filtro definito su misura per gli oggetti in base a prefissi, suffissi, tag dell'oggetto, dimensioni dell'oggetto, età dell'oggetto o una combinazione di questi filtri.

È possibile utilizzare i gruppi S3 Storage Lens per ottenere una visibilità granulare su grandi bucket condivisi, come i data lake, per prendere decisioni aziendali più informate. Ad esempio, è possibile semplificare l'allocazione dello spazio di archiviazione e ottimizzare il reporting dei costi frazionando l'utilizzo dell'archiviazione in gruppi specifici di oggetti per singoli progetti e centri di costo all'interno di un bucket o tra più bucket.

Per utilizzare i gruppi S3 Storage Lens, è necessario aggiornare il pannello di controllo in modo che le raccomandazioni e i parametri avanzati siano accessibili. Per ulteriori informazioni sui gruppi S3 Storage Lens, consulta the section called “Utilizzo dei gruppi S3 Storage Lens”.

Aggiornamento di un pannello di controllo di Amazon S3 Storage Lens

Attieniti alla procedura seguente per aggiornare un pannello di controllo Amazon S3 Storage Lens sulla console Amazon S3.

Fase 1: aggiornare l'ambito del pannello di controllo

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Storage Lens, Pannelli di controllo).
3. Scegli il pannello di controllo che desideri modificare, quindi seleziona Edit (Modifica).

   Viene visualizzata la pagina Edit dashboard (Modifica pannello di controllo).

   > **Note**
   >
   > Non è possibile modificare quanto segue:
   >
   > - Il nome del pannello di controllo
   > - La regione di origine
   > - L'ambito del pannello di controllo predefinito, che fa riferimento all'archiviazione del tuo account nel suo complesso.

4. (Facoltativo) Nella pagina di configurazione del pannello di controllo, nella sezione General (Generale) aggiorna e aggiungi tag al pannello di controllo.

   I tag possono essere utilizzati per gestire le autorizzazioni per il pannello di controllo e tenere traccia dei costi per S3 Storage Lens. Per ulteriori informazioni, consulta Controllo dell'accesso mediante i tag di risorse nella Guida per l'utente di IAM e Tag per l'allocazione dei costi generati da AWS nella Guida per l'utente di AWS Billing .

   > **Note**
   >
   > Puoi aggiungere fino a 50 tag alla configurazione del pannello di controllo.

5. Nella sezione Ambito del pannello di controllo completa le seguenti operazioni:

   a. Aggiorna le regioni e i bucket che desideri siano incluse o escluse da S3 Storage Lens nel pannello di controllo.

      > **Note**
      >
      > - Puoi anche includere o escludere regioni e e bucket. Questa opzione è limitata alle regioni solo se si creano pannelli di controllo a livello di organizzazione tra gli account membri dell'organizzazione.
      > - Puoi scegliere fino a 50 bucket da includere o escludere.

   b. Aggiorna i bucket nelle regioni selezionate che desideri siano inclusi o esclusi da S3 Storage Lens. Puoi includere o escludere i bucket, ma non puoi eseguire entrambe le operazioni. Questa opzione non è disponibile quando si creano pannelli di controllo a livello di organizzazione.

Fase 2: aggiornare la selezione dei parametri

1. Nella sezione Selezione parametri seleziona il tipo parametri che desideri aggregare per il pannello di controllo.

   - Per includere i parametri gratuiti aggregati a livello di bucket e disponibili per le query per 14 giorni, scegli Free metrics (Parametri gratuiti).
   - Per abilitare i parametri avanzati e altre opzioni avanzate, scegli Advanced metrics and recommendations (Parametri e suggerimenti avanzati). Queste opzioni includono

l'aggregazione avanzata dei prefissi, la CloudWatch pubblicazione su Amazon e i consigli contestuali. I dati sono disponibili per le query per 15 mesi. I parametri e i suggerimenti avanzati hanno un costo aggiuntivo. Per ulteriori informazioni, consulta Prezzi di Amazon S3.

Per ulteriori informazioni su parametri avanzati e parametri gratuiti, consulta Selezione dei parametri.

2. In Advanced metrics and recommendations features (Parametri avanzati e funzioni di suggerimento), seleziona le opzioni da abilitare:

   - Advanced metrics (Parametri avanzati)
   - CloudWatch pubblicazione
   - Aggregazione di prefisso

   > **Important**
   >
   > Se abiliti l'aggregazione dei prefissi per la configurazione di S3 Storage Lens, le metriche a livello di prefisso non verranno pubblicate su. CloudWatch Vengono pubblicate solo le metriche di S3 Storage Lens a livello di bucket, account e organizzazione. CloudWatch

3. Se hai abilitato Advanced metrics (Parametri avanzati), in Advanced metrics categories (Categorie parametri avanzati) seleziona le categorie che desideri visualizzare nel pannello di controllo di S3 Storage Lens:

   - Parametri delle attività
   - Detailed status code metrics (Parametri dettagliati codice di stato)
   - Advanced cost optimization metrics (Parametri avanzati ottimizzazione costi)
   - Advanced data protection metrics (Parametri avanzati protezione dati)

   Per ulteriori informazioni sulle categorie di parametri, consulta Categorie di parametri. Per un elenco completo di parametri, consulta Glossario dei parametri di Amazon S3 Storage Lens.

4. Se hai scelto di abilitare l'aggregazione dei prefissi, configura quanto segue:

   a. Scegli la dimensione minima della soglia del prefisso per questo pannello di controllo.

      Ad esempio, una soglia di prefisso del 5% indica che verranno aggregati i prefissi che costituiscono una dimensione pari o superiore al 5% dell'archiviazione del bucket.

b. Dovrai scegliere anche la profondità del prefisso.

   Questa impostazione indica il numero massimo di livelli fino a cui vengono valutati i prefissi. La profondità del prefisso deve essere inferiore a 10.

c. Specifica un carattere delimitatore per il prefisso.

   Questo è il valore utilizzato per identificare ogni livello di prefisso. Il valore predefinito in Amazon S3 è il carattere `/`, ma la struttura dell'archiviazione potrebbe utilizzare altri caratteri delimitatori.

(Facoltativo) Fase 3: esportare i parametri per il pannello di controllo

1. Nella sezione Metrics export (Esportazione parametri), per creare un'esportazione dei parametri che verrà inserita quotidianamente in un bucket di destinazione a tua scelta scegli Enable (Abilita). Per disabilitare l'esportazione dei parametri, scegli Disable (Disabilita).

   I parametri vengono esportati in formato CSV o Apache Parquet. Essa rappresenta lo stesso ambito di dati dei dati del pannello di controllo di S3 Storage Lens senza le raccomandazioni.

2. Se abilitata, scegli il formato di output dell'esportazione giornaliera dei parametri: CSV o Apache Parquet.

   Parquet è un formato di file open source per Hadoop che memorizza i dati nidificati in un formato a colonne semplice.

3. Scegli il bucket S3 di destinazione per l'esportazione dei parametri.

   Puoi scegliere un bucket nell'account corrente del pannello di controllo di S3 Storage Lens. Oppure puoi sceglierne un altro Account AWS se disponi delle autorizzazioni per il bucket di destinazione e dell'ID account del proprietario del bucket di destinazione.

4. Scegli il bucket S3 di destinazione (formato: `s3://`*`bucket-name`*`/`*`prefix`*).

   Il bucket deve trovarsi nella regione principale del pannello di controllo di S3 Storage Lens. Nella casella Destination bucket permission (Autorizzazione bucket di destinazione) della console S3 verrà visualizzata l'autorizzazione che verrà aggiunta da Amazon S3 alla policy di bucket di destinazione. Amazon S3 aggiornerà la policy relative ai bucket sul bucket di destinazione per consentire a S3 di inserire i dati in quel bucket.

5. (Facoltativo) Per abilitare la crittografia lato server per l'esportazione dei parametri, segli Specify an encryption key (Specifica una chiave di crittografia). Quindi scegli il Tipo di crittografia: Chiavi gestite da Amazon S3 (SSE-S3) o Chiave AWS Key Management Service (SSE-KMS).

   Puoi scegliere una chiave gestita da Amazon S3 (SSE-S3) o una chiave AWS Key Management Service (AWS KMS) (SSE-KMS).

6. (Facoltativo) Per specificare una AWS KMS chiave, devi scegliere una chiave KMS o inserire una chiave Amazon Resource Name (ARN). In Chiave AWS KMS specifica la tua chiave KMS in uno dei seguenti modi:

   - Per effettuare una selezione in un elenco di chiavi KMS disponibili, seleziona Scegli tra le chiavi AWS KMS keys e quindi scegli una chiave KMS dell'elenco delle chiavi disponibili.

     In questo elenco vengono visualizzate sia la chiave Chiave gestita da AWS (`aws/s3`) che quella gestita dai clienti. Per ulteriori informazioni sulle chiavi gestite dal cliente, consulta Chiavi gestite dal cliente e chiavi AWS nella Guida per gli sviluppatori di AWS Key Management Service .

     > **Note**
     >
     > Chiave gestita da AWS (`aws/S3`) non è supportato per la crittografia SSE-KMS con S3 Storage Lens.

   - Per specificare l'ARN della chiave KMS, scegli Inserisci l'ARN della AWS KMS key e quindi specifica l'ARN della chiave KMS nel campo visualizzato.
   - Per creare una nuova chiave gestita dal cliente nella AWS KMS console, scegli Crea una chiave KMS.

     Se scegli una chiave gestita dal cliente, devi concedere a S3 Storage Lens l'autorizzazione a eseguire la crittografia nella policy della chiave AWS KMS . Per ulteriori informazioni, consulta Utilizzo di un file AWS KMS key per crittografare le esportazioni delle metriche.

     Per ulteriori informazioni sulla creazione di una AWS KMS key, consulta Creating Keys nella AWS Key Management Service Developer Guide.

7. Seleziona Salvataggio delle modifiche.

Per ottenere ulteriore visibilità sull'archiviazione, è possibile creare uno o più gruppi S3 Storage Lens e collegarli al pannello di controllo. Un gruppo S3 Storage Lens è un filtro definito su misura per gli

oggetti in base a prefissi, suffissi, tag dell'oggetto, dimensioni dell'oggetto, età dell'oggetto o una combinazione di questi filtri.

È possibile utilizzare i gruppi S3 Storage Lens per ottenere una visibilità granulare su grandi bucket condivisi, come i data lake, per prendere decisioni aziendali più informate. Ad esempio, è possibile semplificare l'allocazione dello spazio di archiviazione e ottimizzare il reporting dei costi frazionando l'utilizzo dell'archiviazione in gruppi specifici di oggetti per singoli progetti e centri di costo all'interno di un bucket o tra più bucket.

Per utilizzare i gruppi S3 Storage Lens, è necessario aggiornare il pannello di controllo in modo che le raccomandazioni e i parametri avanzati siano accessibili. Per ulteriori informazioni sui gruppi S3 Storage Lens, consulta the section called “Utilizzo dei gruppi S3 Storage Lens”.

Disabilitazione o eliminazione di un pannello di controllo di Amazon S3 Storage Lens

S3 Storage Lens aggrega i tuoi parametri e mostra le informazioni nella sezione Account snapshot (Snapshot dell'account) nella pagina Buckets (Bucket) della console Amazon S3. S3 Storage Lens dispone inoltre di un pannello di controllo interattivo che puoi usare per visualizzare informazioni dettagliate e tendenze, contrassegnare le anomalie e ricevere consigli per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati. Nel pannello di controllo sono disponibili opzioni di drill-down per generare e visualizzare approfondimenti a livello di organizzazione, account, Regione AWS, classe di archiviazione, bucket, prefisso o gruppo Storage Lens. Puoi anche inviare un'esportazione dei parametri quotidiana in formato CSV o Parquet a un bucket S3.

La dashboard predefinita di Amazon S3 Storage Lens è. default-account-dashboard Questo pannello di controllo è preconfigurato da Amazon S3 per aiutarti a visualizzare informazioni dettagliate di riepilogo e tendenze per i parametri avanzati e gratuiti aggregati dell'intero account nella console. Non puoi modificare l'ambito di configurazione del pannello di controllo predefinito, ma puoi aggiornare la selezione dei parametri dai parametri gratuiti ai suggerimenti e ai parametri avanzati a pagamento, configurare l'esportazione facoltativa dei parametri o addirittura disabilitare il pannello di controllo predefinito. Il pannello di controllo predefinito non può essere eliminato.

Puoi eliminare o disattivare un pannello di controllo di Amazon S3 Storage Lens dalla console Amazon S3. La disabilitazione o l'eliminazione di un pannello di controllo impedisce la generazione di parametri in futuro. Un pannello di controllo disabilitato conserva ancora le informazioni di configurazione, in modo che possano essere facilmente richiamate in caso di riattivazione. Un pannello di controllo disabilitato conserva i dati della cronologia fino a quando non sarà più disponibile per le query.

I dati per le selezioni dei parametri gratuiti sono disponibili per le query per 14 giorni, mentre i dati per le selezioni di suggerimenti e parametri avanzati sono disponibili per le query per 15 mesi.

Disabilitazione di un pannello di controllo di Amazon S3 Storage Lens

Per disabilitare un pannello di controllo di S3 Storage Lens

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Pannello di controllo seleziona il pannello di controllo che desideri disabilitare, quindi seleziona Disabilita nella parte superiore dell'elenco.
4. Nella pagina visualizzata, conferma che desideri realmente disabilitare il pannello di controllo specificando il nome del pannello di controllo nel campo di testo, quindi seleziona Conferma.

Eliminazione di un pannello di controllo di Amazon S3 Storage Lens

> **Note**
>
> Non puoi eliminare il pannello di controllo predefinito. Tuttavia, è possibile effettuarne la disabilitazione. Prima di eliminare un pannello di controllo che hai creato in precedenza, considera quanto segue:
>
> - In alternativa all'eliminazione di un pannello di controllo, puoi disabilitarlo in modo che sia disponibile per una riattivazione in futuro. Per ulteriori informazioni, consulta Disabilitazione di un pannello di controllo di Amazon S3 Storage Lens.
> - L'eliminazione del pannello di controllo comporta l'eliminazione di tutte le impostazioni di configurazione ad esso associate.
> - I dati delle metriche della cronologia non saranno più disponibili. Questi dati storici sono ancora conservati per 15 mesi. Se desideri accedere nuovamente a questi dati, dovrai creare un pannello di controllo con lo stesso nome nella stessa regione di origine di quella eliminata.

Per eliminare un pannello di controllo di S3 Storage Lens

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

2. Nel riquadro di navigazione a sinistra, scegli Storage Lens, Dashboards (Pannelli di controllo).
3. Nell'elenco Pannello di controllo seleziona il pannello di controllo che desideri eliminare, quindi seleziona Elimina nella parte superiore dell'elenco.
4. Nella pagina Elimina pannelli di controllo conferma di voler realmente eliminare il pannello di controllo specificandone il nome nel campo di testo. Quindi scegli Conferma.

Collaborazione con la creazione AWS Organizations di dashboard a livello di organizzazione

S3 Storage Lens aggrega i tuoi parametri e mostra le informazioni nella sezione Account snapshot (Snapshot dell'account) nella pagina Buckets (Bucket) della console di Amazon S3. S3 Storage Lens dispone inoltre di un pannello di controllo interattivo che puoi usare per visualizzare informazioni dettagliate e tendenze, contrassegnare le anomalie e ricevere consigli per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati. Nel pannello di controllo sono disponibili opzioni di drill-down per generare e visualizzare approfondimenti a livello di organizzazione, account, Regione AWS, classe di archiviazione, bucket, prefisso o gruppo Storage Lens. Puoi anche inviare un'esportazione dei parametri quotidiana in formato CSV o Parquet a un bucket S3.

La dashboard predefinita di Amazon S3 Storage Lens è. default-account-dashboard Questo pannello di controllo è preconfigurato da Amazon S3 per aiutarti a visualizzare informazioni dettagliate di riepilogo e tendenze per i parametri avanzati e gratuiti aggregati dell'intero account nella console. Non puoi modificare l'ambito di configurazione del pannello di controllo predefinito, ma puoi aggiornare la selezione dei parametri dai parametri gratuiti ai suggerimenti e ai parametri avanzati a pagamento, configurare l'esportazione facoltativa dei parametri o addirittura disabilitare il pannello di controllo predefinito. Il pannello di controllo predefinito non può essere eliminato.

Puoi anche creare dashboard S3 Storage Lens aggiuntivi incentrati su bucket S3 specifici Regioni AWS o altro all'interno dell'organizzazione. Account AWS

Il pannello di controllo di S3 Storage Lens fornisce numerose informazioni sull'ambito dell'archiviazione. Un pannello di controllo visualizza più di 30 parametri che rappresentano tendenze e informazioni in ambiti quali il riepilogo dello stato dell'archiviazione, l'efficienza dei costi, la protezione dei dati e l'attività.

Amazon S3 Storage Lens può essere utilizzato per raccogliere parametri di storage e dati di utilizzo per tutti gli account che fanno parte della tua gerarchia. AWS Organizations A tale scopo, devi utilizzare AWS Organizations e abilitare l'accesso affidabile di S3 Storage Lens utilizzando il tuo account di gestione. AWS Organizations

Se l'accesso attendibile è abilitato, potrai aggiungere l'accesso da amministratore delegato agli account dell'organizzazione. Questi account possono quindi creare pannelli di controllo e configurazioni a livello di organizzazione per S3 Storage Lens. Per maggiori informazioni sull'abilitazione dell'accesso attendibile, consulta Amazon S3 Lens e AWS Organizations nella Guida per l'utente di AWS Organizations .

I seguenti controlli della console sono disponibili solo per gli account di AWS Organizations gestione.

Abilitazione dell'accesso attendibile per S3 Storage Lens nell'organizzazione

L'abilitazione dell'accesso affidabile consente ad Amazon S3 Storage Lens di accedere alla AWS Organizations gerarchia, all'appartenenza e alla struttura tramite AWS Organizations operazioni API. S3 Storage Lens diventa in questo modo un servizio attendibile per l'intera struttura dell'organizzazione. Può creare ruoli collegati ai servizi negli account di gestione o amministratore delegato dell'organizzazione ogni volta che viene creata una configurazione del pannello di controllo.

Il ruolo collegato ai servizi concede le autorizzazioni S3 Storage Lens per descrivere le organizzazioni, elencare gli account, verificare un elenco di accesso ai servizi per le organizzazioni e ottenere amministratori delegati per l'organizzazione. Ciò consente a S3 Storage Lens di raccogliere metriche di utilizzo e attività dell'archiviazione multi-account per i pannelli di controllo all'interno degli account nell'organizzazione.

Per ulteriori informazioni, consulta Utilizzo dei ruoli collegati ai servizi per Amazon S3 Storage Lens.

> **Note**
>
> - L'accesso attendibile può essere abilitato solo dall'account di gestione.
> - Solo l'account di gestione e gli amministratori delegati possono creare pannelli di controllo o configurazioni di S3 Storage Lens per l'organizzazione.

Per abilitare S3 Storage Lens in modo che abbia un accesso attendibile

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a subustra, seleziona Storage Lens, Organization settings (Impostazioni organizzazione).
3. In Accesso organizzazioni, seleziona Modifica.

Sarà visualizzata la pagina Accesso organizzazione. Da qui puoi abilitare l'accesso attendibile per S3 Storage Lens. Ciò consente all'utente e a tutti gli altri titolari di account aggiunti come amministratori delegati di creare pannelli di controllo per tutti gli account e lo storage dell'organizzazione.

Disabilitazione dell'accesso attendibile di S3 Storage Lens nell'organizzazione

Disabilitando l'accesso attendibile, si limita S3 Storage Lens al funzionamento solo a livello di account. Ogni titolare dell'account può visualizzare i vantaggi di S3 Storage Lens limitati all'ambito del proprio account e non all'intera organizzazione. Tutti i pannelli di controllo che richiedono un accesso attendibile non saranno più aggiornati, ma potranno eseguire query sui dati della cronologia in base ai rispettivi periodi di disponibilità dei dati per le query.

La rimozione di un account come amministratore delegato limiterà l'accesso alle metriche del pannello di controllo di S3 Storage Lens del proprietario dell'account in modo da funzionare solo a livello di account. I pannelli di controllo organizzativi creati non saranno più aggiornati, ma conserveranno i dati della cronologia in base ai rispettivi periodi di conservazione per query.

> **Note**
>
> - La disabilitazione dell'accesso attendibile disabilita automaticamente tutti i pannelli di controllo a livello di organizzazione, poiché S3 Storage Lens non avrà più accesso attendibile agli account dell'organizzazione per raccogliere e aggregare i parametri di storage.
> - Gli account di gestione e amministratore delegato possono ancora visualizzare i dati della cronologia per questi pannelli di controllo disabilitati e possono eseguire query su questi dati mentre sono disponibili.

Per disabilitare l'accesso attendibile per S3 Storage Lens

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a subustra, seleziona Storage Lens, Organization settings (Impostazioni organizzazione).
3. In Accesso organizzazioni, seleziona Modifica.

Sarà visualizzata la pagina Accesso organizzazione. Da qui puoi disabilitare l'accesso attendibile per S3 Storage Lens.

Registrazione di amministratori delegati per S3 Storage Lens

Dopo aver abilitato l'accesso attendibile, potrai registrare l'accesso come amministratore delegato agli account dell'organizzazione. Quando un account viene registrato come amministratore delegato, l'account riceve l'autorizzazione ad accedere a tutte le operazioni API di sola lettura AWS Organizations . Ciò fornisce visibilità ai membri e alle strutture dell'organizzazione in modo che possano creare pannelli di controllo di S3 Storage Lens per conto dell'utente.

Per registrare gli amministratori delegati per S3 Storage Lens

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a subustra, seleziona Storage Lens, Organization settings (Impostazioni organizzazione).
3. Nella sezione Accesso delegato, in Account, seleziona Aggiungi account.

   Viene visualizzata la pagina Accesso amministratore delegato. Da qui potrai aggiungere un ID Account AWS come amministratore delegato per creare pannelli di controllo a livello di organizzazione per tutti gli account e l'archiviazione dell'organizzazione.

Annullamento della registrazione di amministratori delegati per S3 Storage Lens

Puoi annullare la registrazione dell'accesso amministratore delegato agli account dell'organizzazione. Quando un account viene annullato come amministratore delegato, l'account perde l'autorizzazione ad accedere a tutte le operazioni AWS Organizations API di sola lettura che forniscono visibilità ai membri e alle strutture dell'organizzazione.

> **Note**
>
> - L'annullamento della registrazione di un amministratore delegato disabilita automaticamente tutti i pannelli di controllo a livello di organizzazione creati dall'amministratore delegato.

- Gli account amministratore delegato potranno ancora visualizzare i dati della cronologia per questi pannelli di controllo disabilitati in base ai rispettivi periodi di conservazione.

Per annullare la registrazione degli account per l'accesso amministratore delegato

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a subustra, seleziona Storage Lens, Organization settings (Impostazioni organizzazione).
3. Nella sezione Account con accesso delegato scegli l'ID account per cui desideri annullare la registrazione, quindi seleziona Rimuovi.

Esempi di Amazon S3 Storage Lens che utilizzano la AWS CLI

S3 Storage Lens aggrega i tuoi parametri e mostra le informazioni nella sezione Account snapshot (Snapshot dell'account) nella pagina Buckets (Bucket) della console Amazon S3. S3 Storage Lens dispone inoltre di un pannello di controllo interattivo che puoi usare per visualizzare informazioni dettagliate e tendenze, contrassegnare le anomalie e ricevere consigli per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati. Nel pannello di controllo sono disponibili opzioni di drill-down per generare e visualizzare approfondimenti a livello di organizzazione, account, Regione AWS, classe di storage, bucket, prefisso o gruppo Storage Lens. Puoi anche inviare un'esportazione dei parametri quotidiana in formato CSV o Parquet a un bucket S3. Per ulteriori informazioni, consulta Valutazione dell'attività e dell'utilizzo dello storage con Amazon S3 Storage Lens.

Di seguito sono riportati esempi che mostrano come utilizzare S3 Storage Lens con la AWS Command Line Interface.

Argomenti

- File helper per l'utilizzo di Amazon S3 Storage Lens
- Utilizzo delle configurazioni di Amazon S3 Storage Lens con la AWS CLI
- Utilizzo di Amazon S3 Storage Lens con esempi di AWS Organizations utilizzando la AWS CLI

File helper per l'utilizzo di Amazon S3 Storage Lens

Utilizza i seguenti file JSON e i suoi input chiave per i tuoi esempi.

Esempio di configurazione di S3 Storage Lens in JSON

Example **`config.json`**

Il file `config.json` contiene i dettagli relativi alla configurazione di parametri e suggerimenti avanzati a livello di organizzazioni S3 Storage Lens. Per utilizzare il seguente esempio, sostituisci *`user input placeholders`* con le tue informazioni.

> **Note**
>
> Per i suggerimenti e i parametri avanzati verranno applicati costi aggiuntivi. Per ulteriori informazioni, consulta Parametri avanzati e suggerimenti.

```
{
  "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
    }
  }
}
```

Esempio di configurazione di S3 Storage Lens con gruppi Storage Lens in JSON

Example **`config.json`**

Il file `config.json` contiene i dettagli da applicare alla configurazione di Storage Lens quando si usano i gruppi Storage Lens. Per usare questo esempio, sostituisci *`user input placeholders`* con le tue informazioni.

Per collegare tutti i gruppi Storage Lens al pannello di controllo, aggiorna la configurazione di Storage Lens con la seguente sintassi:

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
      "IsEnabled":true
      },
    "StorageLensGroupLevel": {},
  "IsEnabled": true
}
```

Per includere solo due gruppi Storage Lens nella configurazione del pannello di controllo Storage Lens (*`slg-1`* e *`slg-2`*), usa la seguente sintassi:

```
{
  "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
}
```

Per escludere solo alcuni gruppi Storage Lens dalla configurazione del pannello di controllo, utilizza la seguente sintassi:

```
{
  "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
}
```

File JSON con tag per configurazione di esempio di S3 Storage Lens

Example **`tags.json`**

Il file `tags.json` contiene i tag da applicare alla configurazione di S3 Storage Lens. Per utilizzare questo comando, sostituisci *`user input placeholders`* con le tue informazioni.

```
[
    {
        "Key": "key1",
        "Value": "value1"
    },
    {
        "Key": "key2",
        "Value": "value2"
    }
]
```

Esempio di configurazione delle autorizzazioni IAM di S3 Storage Lens

Example **`permissions.json`**: nome del pannello di controllo specifico

Questa policy di esempio mostra il file `permissions.json` IAM di S3 Storage Lens con un nome specificato per il pannello di controllo. Sostituisci *`value1`*, *`us-east-1`*, *`your-dashboard-name`* e *`example-account-id`* con il tuo valore.

```
{
    "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`**: nessun nome del pannello di controllo specifico

Questa policy di esempio mostra il file `permissions.json` IAM di S3 Storage Lens senza un nome specificato per il pannello di controllo. Sostituisci *`value1`*, *`us-east-1`* e *`example-account-id`* con il tuo valore.

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

Utilizzo delle configurazioni di Amazon S3 Storage Lens con la AWS CLI

Puoi utilizzare la AWS CLI per visualizzare, creare, eliminare, ottenere, contrassegnare e aggiornare le configurazioni di S3 Storage Lens. Negli esempi seguenti vengono utilizzati i file JSON helper per gli input chiave. Per usare questi esempi, sostituisci *`user input placeholders`* con le tue informazioni.

Creare e aggiornare una configurazione di S3 Storage Lens

Example Creare e aggiornare una configurazione di S3 Storage Lens

```
aws s3control put-storage-lens-configuration --account-id=111122223333 --
config-id=example-dashboard-configuration-id --region=us-east-1 --storage-lens-
configuration=file://./config.json --tags=file://./tags.json
```

Creare e aggiornare una configurazione di S3 Storage Lens senza tag

Example Creare e aggiornare una configurazione di S3 Storage Lens senza tag

```
aws s3control put-storage-lens-configuration --account-id=222222222222 --config-
id=your-configuration-id --region=us-east-1 --storage-lens-configuration=file://./
config.json
```

Ottieni una configurazione di S3 Storage Lens

Example Ottieni una configurazione di S3 Storage Lens

```
aws s3control get-storage-lens-configuration --account-id=222222222222 --config-
id=your-configuration-id --region=us-east-1
```

Visualizzare le configurazioni di S3 Storage Lens senza il token successivo

Example Visualizzare le configurazioni di S3 Storage Lens senza il token successivo

```
aws s3control list-storage-lens-configurations --account-id=222222222222 --region=us-
east-1
```

Visualizza le configurazioni di S3 Storage Lens

Example Visualizza le configurazioni di S3 Storage Lens

```
aws s3control list-storage-lens-configurations --account-id=222222222222 --region=us-
east-1 --next-token=abcdefghij1234
```

Elimina una configurazione di S3 Storage Lens

Example Elimina una configurazione di S3 Storage Lens

```
aws s3control delete-storage-lens-configuration --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id
```

Aggiungere tag a una configurazione di S3 Storage Lens

Example Aggiungere tag a una configurazione di S3 Storage Lens

```
aws s3control put-storage-lens-configuration-tagging --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id --tags=file://./tags.json
```

Ottieni i tag per una configurazione di S3 Storage Lens

Example Ottieni i tag per una configurazione di S3 Storage Lens

```
aws s3control get-storage-lens-configuration-tagging --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id
```

Elimina i tag per una configurazione di S3 Storage Lens

Example Elimina i tag per una configurazione di S3 Storage Lens

```
aws s3control delete-storage-lens-configuration-tagging --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id
```

Utilizzo di Amazon S3 Storage Lens con esempi di AWS Organizations utilizzando la AWS CLI

Utilizza Amazon S3 Storage Lens per raccogliere i parametri di archiviazione e i dati di utilizzo per tutti gli account che fanno parte della tua gerarchia di AWS Organizations. Per maggiori informazioni, consulta Utilizzo di Amazon S3 Storage Lens con AWS Organizations.

Abilita l'accesso attendibile di Organizations per S3 Storage Lens

Example Abilita l'accesso attendibile di Organizations per S3 Storage Lens

```
aws organizations enable-aws-service-access --service-principal storage-lens.s3.amazonaws.com
```

Disabilita l'accesso attendibile di Organizations per S3 Storage Lens

Example Disabilita l'accesso attendibile di Organizations per S3 Storage Lens

```
aws organizations disable-aws-service-access --service-principal storage-
lens.s3.amazonaws.com
```

Registra gli amministratori delegati di Organizations per S3 Storage Lens

Example Registra gli amministratori delegati di Organizations per S3 Storage Lens

Per utilizzare questo esempio, sostituisci *111122223333* con l'ID Account AWS appropriato.

```
aws organizations register-delegated-administrator --service-principal storage-
lens.s3.amazonaws.com --account-id 111122223333
```

Cancella gli amministratori delegati di Organizations per S3 Storage Lens

Example Cancella gli amministratori delegati di Organizations per S3 Storage Lens

Per utilizzare questo esempio, sostituisci *111122223333* con l'ID Account AWS appropriato.

```
aws organizations deregister-delegated-administrator --service-principal storage-
lens.s3.amazonaws.com --account-id 111122223333
```

Esempi di Amazon S3 Storage Lens che utilizzano l'SDK per Java

S3 Storage Lens aggrega i tuoi parametri e mostra le informazioni nella sezione Account snapshot (Snapshot dell'account) nella pagina Buckets (Bucket) della console Amazon S3. S3 Storage Lens dispone inoltre di un pannello di controllo interattivo che puoi usare per visualizzare informazioni dettagliate e tendenze, contrassegnare le anomalie e ricevere consigli per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati. Nel pannello di controllo sono disponibili opzioni di drill-down per generare e visualizzare approfondimenti a livello di organizzazione, account, Regione AWS, classe di storage, bucket, prefisso o gruppo Storage Lens. Puoi anche inviare un'esportazione dei parametri quotidiana in formato CSV o Parquet a un bucket S3. Per ulteriori informazioni, consulta Valutazione dell'attività e dell'utilizzo dello storage con Amazon S3 Storage Lens.

Di seguito sono riportati esempi che mostrano come utilizzare S3 Storage Lens con la AWS SDK for Java.

Argomenti

Utilizzo delle configurazioni di Amazon S3 Storage Lens tramite l'SDK per Java

Puoi utilizzare l'SDK per Java per visualizzare, creare, ottenere e aggiornare le configurazioni di S3 Storage Lens. Negli esempi seguenti vengono utilizzati i file JSON helper per gli input chiave.

Argomenti

Crea e aggiorna una configurazione di S3 Storage Lens

Example Crea e aggiorna una configurazione di 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();
        }
    }
}
```

Elimina una configurazione di S3 Storage Lens

Example Elimina una configurazione di 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();
        }
    }
}
```

Ottieni una configurazione di S3 Storage Lens

Example Ottieni una configurazione di 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();
        }
    }
}
```

Visualizza le configurazioni di S3 Storage Lens

Example Visualizza le configurazioni di 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();
        }
    }
}
```

Aggiungere tag a una configurazione di S3 Storage Lens

Example Aggiungere tag a una configurazione di 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();
        }
    }
}
```

Ottieni i tag per una configurazione di S3 Storage Lens

Example Ottieni i tag per una configurazione di 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();
        }
    }
}
```

Elimina i tag per una configurazione di S3 Storage Lens

Example Elimina i tag per una configurazione di 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();
        }
    }
}
```

Aggiornare la configurazione predefinita di S3 Storage Lens con parametri e suggerimenti avanzati

Example Aggiornare la configurazione predefinita di S3 Storage Lens con parametri avanzati e suggerimenti

```
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**
>
> Per i suggerimenti e i parametri avanzati verranno applicati costi aggiuntivi. Per ulteriori informazioni, consulta Parametri avanzati e suggerimenti.

Collegare un gruppo Storage Lens a un pannello di controllo S3 Storage Lens

Example Collegare tutti i gruppi Storage Lens a un pannello di controllo

Nel seguente esempio di SDK per Java tutti i gruppi Storage Lens nell'account *111122223333* vengono collegati al pannello di controllo *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 Collegare due gruppi Storage Lens a un pannello di controllo

Nel seguente esempio di AWS SDK for Java due gruppi Storage Lens (*StorageLensGroupName1* e *StorageLensGroupName2*) vengono collegati al pannello di controllo *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 Collegare tutti i gruppi Storage Lens con esclusioni

Nel seguente esempio di SDK per Java tutti i gruppi Storage Lens al pannello di controllo *ExampleDashboardConfigurationId*, ad esclusione dei due gruppi specificati (*StorageLensGroupName1* e *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();
        }
    }
}
```

Utilizzo di Amazon S3 Storage Lens con esempi di AWS Organizations tramite l'SDK per Java

Utilizza Amazon S3 Storage Lens per raccogliere i parametri di archiviazione e i dati di utilizzo per tutti gli account che fanno parte della tua gerarchia di AWS Organizations. Per maggiori informazioni, consulta Utilizzo di Amazon S3 Storage Lens con AWS Organizations.

Argomenti

- Abilita l'accesso attendibile di Organizations per S3 Storage Lens
- Disabilita l'accesso attendibile di Organizations per S3 Storage Lens
- Registra gli amministratori delegati di Organizations per S3 Storage Lens
- Cancella gli amministratori delegati di Organizations per S3 Storage Lens

Abilita l'accesso attendibile di Organizations per S3 Storage Lens

Example Abilita l'accesso attendibile di Organizations per 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();
        }
 }
}
```

Disabilita l'accesso attendibile di Organizations per S3 Storage Lens

Example Disabilita l'accesso attendibile di Organizations per 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();
        }
 }
}
```

Registra gli amministratori delegati di Organizations per S3 Storage Lens

Example Registra gli amministratori delegati di Organizations per 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();
        }
 }
}
```

Cancella gli amministratori delegati di Organizations per S3 Storage Lens

Example Cancella gli amministratori delegati di Organizations per 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();
        }
 }
}
```

Utilizzo dei gruppi S3 Storage Lens

Un gruppo Amazon S3 Storage Lens aggrega i parametri utilizzando filtri personalizzati basati sui metadati degli oggetti. I gruppi Storage Lens ti aiutano ad approfondire le caratteristiche dei tuoi dati, come la distribuzione degli oggetti per età, i tipi di file più comuni e altro ancora. Ad esempio, puoi filtrare le metriche per tag di oggetto per identificare i set di dati in più rapida crescita o visualizzare lo storage in base alla dimensione e all'età degli oggetti per definire la tua strategia di archiviazione dello storage. Di conseguenza, i gruppi Amazon S3 Storage Lens ti aiutano a comprendere e ottimizzare meglio la tua archiviazione S3.

Con i gruppi Storage Lens, puoi analizzare e filtrare i parametri di S3 Storage Lens utilizzando metadati degli oggetti come prefissi, suffissi, tag degli oggetti, dimensioni o età degli oggetti. Puoi anche applicare una combinazione di questi filtri. Dopo aver collegato il gruppo Storage Lens al

pannello di controllo di S3 Storage Lens, puoi visualizzare i parametri S3 Storage Lens aggregati in base ai gruppi Amazon S3 Storage Lens direttamente nel pannello di controllo.

Ad esempio, è anche possibile filtrare i parametri per dimensione degli oggetti o fasce di età per determinare quale parte dell'archiviazione è costituita da oggetti di piccole dimensioni. Quindi puoi utilizzare queste informazioni con S3 Intelligent-Tiering o S3 Lifecycle per trasferire piccoli oggetti in classi di archiviazione diverse per ottimizzare costi e archiviazione.

Argomenti

Funzionamento dei gruppi S3 Storage Lens

I gruppi Storage Lens consentono di aggregare i parametri utilizzando filtri personalizzati basati sui metadati degli oggetti. Quando si definisce un filtro personalizzato, è possibile utilizzare prefissi, suffissi, tag degli oggetti, dimensioni degli oggetti, età degli oggetti o una combinazione di questi filtri personalizzati. Durante la creazione del gruppo Storage Lens, puoi anche includere un singolo filtro o più condizioni di filtro. Per specificare più condizioni di filtro, utilizza gli operatori logici `And` oppure `Or`.

Quando crei e configuri un gruppo Storage Lens, questo opera da filtro personalizzato nel pannello di controllo al quale colleghi il gruppo. Nel pannello di controllo, puoi quindi utilizzare il filtro di gruppo Storage Lens per ottenere parametri di archiviazione basati sul filtro personalizzato che hai definito nel gruppo.

Per visualizzare i dati del gruppo Storage Lens nel pannello di controllo S3 Storage Lens, devi creare il gruppo e poi collegarlo al pannello di controllo. Dopo aver collegato il gruppo Storage Lens al pannello di controllo Storage Lens, quest'ultimo raccoglierà i parametri di utilizzo dell'archiviazione entro 48 ore. Potrai visualizzare questi dati nel pannello di controllo Storage Lens oppure esportarli tramite un'esportazione dei parametri. Se dimentichi di collegare un gruppo Storage Lens a una pannello di controllo, i dati del gruppo Storage Lens non verranno acquisiti e quindi neppure visualizzati.

> **Note**
>
> - Quando crei un gruppo S3 Storage Lens, crei una risorsa AWS. Pertanto, ogni gruppo Storage Lens ha il proprio nome della risorsa Amazon (ARN), che puoi specificare quando lo colleghi o lo escludi da un pannello di controllo S3 Storage Lens.

- Se il tuo gruppo Storage Lens non è collegato a un pannello di controllo, non dovrai sostenere costi aggiuntivi per la creazione di un gruppo Storage Lens.
- S3 Storage Lens aggrega i parametri di utilizzo di un oggetto in tutti i gruppi Storage Lens corrispondenti. Pertanto, se un oggetto soddisfa le condizioni di filtro per due o più gruppi Storage Lens, i conteggi relativi all'utilizzo dell'archiviazione saranno visualizzati come ripetuti per lo stesso oggetto.

È possibile creare un gruppo Storage Lens a livello di account in una regione di origine specificata tra quelle nell'elenco delle Regioni AWS supportate. Quindi, puoi collegare il tuo gruppo Storage Lens a più pannelli di controllo Storage Lens, purché tali pannelli si trovino nello stesso Account AWS e nella stessa regione. È possibile creare fino a 50 gruppi Storage Lens con la stessa regione di origine in ogni Account AWS.

Puoi creare e gestire le configurazioni del pannello di controllo di S3 Storage Lens utilizzando la console Amazon S3, la AWS Command Line Interface (AWS CLI), gli SDK AWS o la REST API di Amazon S3.

Argomenti

Visualizzazione dei parametri aggregati del gruppo Storage Lens

È possibile visualizzare i parametri aggregati per i gruppi Storage Lens collegando i gruppi a un pannello di controllo. I gruppi Storage Lens che desideri collegare devono risiedere nella regione di origine designata nell'account del pannello di controllo.

Per collegare un gruppo Storage Lens a un pannello di controllo, è necessario specificare il gruppo nella sezione Aggregazione dei gruppi Storage Lens della configurazione del pannello di controllo. Se hai più gruppi Storage Lens, puoi filtrare i risultati in Aggregazione dei gruppi Storage Lens per includere o escludere solo i gruppi che desideri. Per ulteriori informazioni su come collegare gruppi ai pannelli di controllo, consulta the section called “Collegare o rimuovere un gruppo Storage Lens”.

Dopo aver collegato i gruppi, i dati di aggregazione dei gruppi di Storage Lens ulteriori saranno visibili nel pannello di controllo entro 48 ore.

> **Note**
>
> Per visualizzare i parametri aggregati relativi al tuo gruppo Storage Lens, devi collegare il gruppo a pannello di controllo S3 Storage Lens.

Autorizzazioni gruppi Storage Lens

I gruppi Storage Lens richiedono autorizzazioni specifiche in AWS Identity and Access Management (IAM) per consentire l'accesso alle azioni del gruppo S3 Storage Lens. Per concedere queste autorizzazioni, puoi utilizzare una policy IAM basata sull'identità. Ti basterà collegare la policy agli utenti, ai gruppi o ai ruoli IAM ai quali devi concedere le autorizzazioni. Tali autorizzazioni possono includere la possibilità di creare o eliminare gruppi Storage Lens, visualizzarne le configurazioni o gestirne i tag.

L'utente o il ruolo IAM a cui concedi le autorizzazioni deve appartenere all'account che ha creato o possiede il gruppo Storage Lens.

Per utilizzare i gruppi Storage Lens e visualizzare i parametri dei gruppi Storage Lens, devi prima disporre delle autorizzazioni appropriate per utilizzare S3 Storage Lens. Per ulteriori informazioni, consulta the section called “Autorizzazioni S3 Storage Lens”.

Per creare e gestire gruppi S3 Storage Lens, devi disporre delle seguenti autorizzazioni IAM, a seconda delle azioni che desideri eseguire:

Azione	Autorizzazioni IAM
Creare un nuovo gruppo Storage Lens	`s3:CreateStorageLensGroup`
Creare un nuovo gruppo Storage Lens con tag	`s3:CreateStorageLensGroup` , `s3:TagResource`
Aggiornare un gruppo Storage Lens esistente	`s3:UpdateStorageLensGroup`
Restituire i dettagli di una configurazione del gruppo Storage Lens	`s3:GetStorageLensGroup`

Azione	Autorizzazioni IAM
Elencare tutti i gruppi Storage Lens nella tua regione di origine	`s3:ListStorageLensGroups`
Eliminare un gruppo Storage Lens	`s3:DeleteStorageLensGroup`
Elencare i tag aggiunti al tuo gruppo Storage Lens	`s3:ListTagsForResource`
Aggiungere o aggiornare un tag di gruppo Storage Lens per un gruppo Storage Lens esistente	`s3:TagResource`
Eliminare un tag da un gruppo Storage Lens	`s3:UntagResource`

Ecco un esempio di come configurare la policy IAM nell'account che crea il gruppo Storage Lens. Per utilizzare questa policy, sostituisci *us-east-1* con la regione di origine in cui si trova il gruppo Storage Lens. Sostituisci *111122223333* con l'ID del tuo Account AWS e sostituisci *example-storage-lens-group* con il nome del gruppo Storage Lens. Per applicare queste autorizzazioni a tutti i gruppi Storage Lens, sostituisci *example-storage-lens-group* con *.

```
{
    "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"
```

```
        }
    ]
}
```

Per ulteriori informazioni sull'utilizzo di S3 Storage Lens, consulta Autorizzazioni Amazon S3 Storage Lens. Per informazioni sul linguaggio delle policy IAM, consulta Politiche e autorizzazioni in Amazon S3.

Configurazione dei gruppi Storage Lens

Nome dei gruppi S3 Storage Lens

Ti consigliamo di assegnare ai gruppi Storage Lens nomi che ne indichino lo scopo, in modo da poter determinare facilmente quali gruppi collegare ai pannelli di controllo. Per collegare un gruppo Storage Lens a un pannello di controllo, è necessario specificare il gruppo nella sezione Aggregazione dei gruppi Storage Lens della configurazione del pannello di controllo.

I nomi dei gruppi Storage Lens devono essere univoci all'interno dell'account. Non devono superare i 64 caratteri e possono contenere solo lettere (a-z, A-Z), numeri (0-9), trattini (-) o trattini bassi (_).

Regione di origine

La regione di origine è la Regione AWS in cui viene creato e gestito il gruppo Storage Lens. Il gruppo Storage Lens viene creato nella stessa regione di origine del pannello di controllo Amazon S3 Storage Lens. Anche la configurazione e i parametri del gruppo Storage Lens vengono archiviati in questa regione. È possibile creare fino a 50 gruppi Storage Lens nella stessa regione di origine.

Dopo aver creato il gruppo Storage Lens, non sarà possibile modificarne la regione.

Ambito

Affinché possano essere inclusi nel gruppo Storage Lens, gli oggetti devono rientrare nell'ambito del pannello di controllo Amazon S3 Storage Lens. L'ambito del pannello di controllo Storage Lens è determinato dai bucket che includi nell'ambito del pannello di controllo della configurazione del pannello di controllo di S3 Storage Lens.

Puoi utilizzare diversi filtri relativi agli oggetti per definire l'ambito del gruppo Storage Lens. Per visualizzare questi parametri del gruppo Storage Lens nel pannello di controllo di S3 Storage Lens, gli oggetti devono corrispondere ai filtri che includi nei gruppi Storage Lens. Ad esempio, supponiamo che il gruppo Storage Lens includa oggetti con il prefisso `marketing` e il suffisso `.png`, ma che

nessun oggetto soddisfi tali criteri. In questo caso, i parametri per questo gruppo Storage Lens non verranno generati nell'esportazione giornaliera dei parametri e nessun parametro per questo gruppo sarà visibile nel pannello di controllo.

Filtri

Puoi utilizzare i seguenti filtri in un gruppo S3 Storage Lens:

- Prefissi: specifica il prefisso degli oggetti inclusi, ovvero una stringa di caratteri all'inizio del nome della chiave dell'oggetto. Ad esempio, il valore `images` per il filtro Prefissi include oggetti con uno dei seguenti prefissi: `images/`, `images-marketing` e `images/production`. La lunghezza massima di un prefisso è 1.024 byte.
- Suffissi: specifica il suffisso degli oggetti inclusi (ad esempio, `.png`, `.jpeg` o `.csv`). La lunghezza massima di un suffisso è 1.024 byte.
- Tag degli oggetti: specifica l'elenco dei tag degli oggetti che desideri filtrare. Una chiave di tag non può superare i 128 caratteri Unicode e il valore di un tag non può superare i 256 caratteri Unicode. Nota che se il campo del valore del tag dell'oggetto viene lasciato vuoto, i gruppi di S3 Storage Lens associano l'oggetto solo ad altri oggetti che hanno anche valori di tag vuoti.
- Età: specifica l'intervallo di età degli oggetti inclusi, che viene espressa in giorni. Sono supportati solo i numeri interi.
- Dimensioni: specifica l'intervallo di dimensioni degli oggetti inclusi, che viene espresso in byte. Sono supportati solo i numeri interi. Il valore massimo consentito è 5 TB.

Tag degli oggetti del gruppo Storage Lens

È possibile creare un gruppo Storage Lens che includa fino a 10 filtri per tag di oggetti. L'esempio seguente include due coppie chiave-valore di tag di oggetto che operano da filtri per un gruppo Storage Lens denominato *`Marketing-Department`*. Per utilizzare questo esempio, sostituisci *`Marketing-Department`* con il nome del gruppo e sostituisci *`object-tag-key-1`*, *`object-tag-value-1`* e così via con le coppie chiave-valore del tag dell'oggetto che desideri filtrare.

```
{
    "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"
                }
            ]
    }
}
```

Operatori logici (**`And`** o **`Or`**)

Per includere più condizioni di filtro nel gruppo Storage Lens, è possibile utilizzare operatori logici (`And` o `Or`). Nell'esempio seguente, il gruppo Storage Lens denominato *`Marketing-Department`* ha un operatore `And` che contiene i filtri `Prefix`, `ObjectAge` e `ObjectSize`. Poiché viene utilizzato un operatore `And`, solo gli oggetti che soddisfano tutte queste condizioni di filtro verranno inclusi nell'ambito del gruppo Storage Lens.

Per utilizzare questo esempio, sostituisci *`user input placeholders`* con i valori in base ai quali desideri filtrare.

```
{
    "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**
>
> Se desideri includere oggetti che soddisfano una qualsiasi delle condizioni di filtro, sostituisci l'operatore logico `And` con l'operatore logico `Or` in questo esempio.

Tag delle risorse AWS

Ogni gruppo S3 Storage Lens viene conteggiato come una risorsa AWS con un proprio nome della risorsa Amazon (ARN). Pertanto, quando configuri il gruppo Storage Lens, puoi aggiungere facoltativamente tag delle risorse AWS al gruppo. È possibile aggiungere fino a 50 tag per ogni gruppo Storage Lens. Per creare un gruppo Storage Lens con tag, devi disporre delle autorizzazioni `s3:CreateStorageLensGroup` e `s3:TagResource`.

È possibile utilizzare i tag delle risorse AWS per classificare le risorse in base al reparto, alla linea di business o al progetto. Ciò è utile quando si dispone di numerose risorse dello stesso tipo. Applicando i tag, è possibile individuare rapidamente un gruppo Storage Lens in base ai tag che hai assegnato loro. È possibile utilizzare i tag anche per monitorare e allocare i costi.

Inoltre, quando aggiungi un tag delle risorse AWS al tuo gruppo Storage Lens, attivi il controllo degli accessi basato su attributi (ABAC). L'ABAC è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi, in questo caso ai tag. È possibile utilizzare le condizioni per specificare i tag delle risorse nelle policy IAM per controllare l'accesso alle risorse AWS.

Puoi modificare chiavi e valori di tag e rimuovere tag da una risorsa in qualsiasi momento. Inoltre, tieni presente le limitazioni seguenti:

- I valori e le chiavi dei tag rispettano la distinzione tra maiuscole e minuscole.
- Se aggiungi un tag con la stessa chiave di un tag esistente a una risorsa specifica, il nuovo valore sovrascrive quello precedente.
- Se elimini una risorsa, verranno eliminati anche tutti i tag associati alla risorsa.
- Non includere dati privati o sensibili nei tag delle risorse AWS.
- I tag di sistema (con chiavi di tag che iniziano con `aws:`) non sono supportati.
- La lunghezza di ogni chiave di tag non può superare i 128 caratteri. La lunghezza di ogni valore di tag non può superare i 256 caratteri.

Esportazione dei parametri dei gruppi Storage Lens

I parametri del gruppo S3 Storage Lens sono inclusi nell'esportazione parametri di Amazon S3 Storage Lens per il pannello di controllo a cui è collegato il gruppo Storage Lens. Per informazioni generali sulla funzionalità di esportazione dei parametri di Storage Lens, consulta Visualizzazione dei parametri di Amazon S3 Storage Lens utilizzando una esportazione di dati.

L'esportazione dei parametri dei gruppi Storage Lens include tutti i parametri di S3 Storage Lens che rientrano nell'ambito del pannello di controllo che hai collegato al gruppo Storage Lens. L'esportazione include anche dati aggiuntivi relativi ai parametri per i gruppi Storage Lens.

Una volta creato il gruppo Storage Lens, l'esportazione dei parametri viene inviata quotidianamente al bucket che hai selezionato al momento di configurare l'esportazione dei parametri per il pannello di controllo a cui è collegato il gruppo. Per ricevere la prima esportazione dei parametri possono essere necessarie fino a 48 ore.

Per generare i parametri dell'esportazione quotidiana, gli oggetti devono corrispondere ai filtri che includi nei gruppi Storage Lens. Se nessun oggetto corrisponde ai filtri che hai incluso nel gruppo Storage Lens, non verrà generato alcun parametro. Tuttavia, se un oggetto corrisponde a due o più gruppi Storage Lens, l'oggetto viene elencato separatamente per ogni gruppo nell'esportazione dei parametri.

Per identificare i parametri per i gruppi Storage Lens cerca uno dei seguenti valori nella colonna `record_type` dell'esportazione dei parametri per il pannello di controllo:

- `STORAGE_LENS_GROUP_BUCKET`
- `STORAGE_LENS_GROUP_ACCOUNT`

La colonna `record_value` mostra l'ARN della risorsa per il gruppo Storage Lens (ad esempio, `arn:aws:s3:us-east-1:111122223333:storage-lens-group/Marketing-Department`).

Utilizzo dei gruppi Storage Lens

I gruppi Amazon S3 Storage Lens aggregano i parametri utilizzando filtri personalizzati basati sui metadati degli oggetti. È possibile analizzare e filtrare i parametri di S3 Storage Lens utilizzando prefissi, suffissi, tag degli oggetti, dimensioni o età degli oggetti. Con i gruppi Amazon S3 Storage Lens puoi anche classificare l'utilizzo all'interno di e tra bucket Amazon S3. Di conseguenza, sarai in grado di comprendere e ottimizzare meglio la tua archiviazione S3.

Per iniziare a visualizzare i dati per un gruppo Storage Lens, devi prima collegare il gruppo Storage Lens a un pannello di controllo di S3 Storage Lens. Se devi gestire i gruppi di Storage Lens nel pannello di controllo, puoi modificare la configurazione del pannello di controllo. Per verificare quali gruppi Storage Lens sono presenti nel tuo account, elencali. Per verificare quali gruppi Storage Lens sono collegati al pannello di controllo, puoi controllare in qualsiasi momento la scheda Gruppi Storage Lens nel pannello di controllo. Accedi ai dettagli di un gruppo Storage Lens esistente per rivedere o aggiornare il suo ambito. È inoltre possibile eliminare definitivamente un gruppo Storage Lens.

Per gestire le autorizzazioni, è possibile creare e aggiungere tag delle risorse AWS definiti dall'utente ai gruppi Storage Lens. È possibile utilizzare i tag delle risorse AWS per classificare le risorse in base al reparto, alla linea di business o al progetto. Ciò è utile quando si dispone di numerose risorse dello stesso tipo. Applicando i tag, è possibile individuare rapidamente un gruppo Storage Lens in base ai tag che hai assegnato loro.

Inoltre, quando aggiungi un tag delle risorse AWS al tuo gruppo Storage Lens, attivi il controllo degli accessi basato su attributi (ABAC). L'ABAC è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi, in questo caso ai tag. È possibile utilizzare le condizioni per specificare i tag delle risorse nelle policy IAM per controllare l'accesso alle risorse AWS.

Argomenti

Creazione di un gruppo Storage Lens

Gli esempi seguenti mostrano come creare un gruppo Amazon S3 Storage Lens utilizzando la console Amazon S3 AWS Command Line Interface ,AWS CLI() e. AWS SDK for Java

Utilizzo della console S3

Per creare un gruppo Storage Lens

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nella barra di navigazione nella parte superiore della pagina, scegli il nome della AWS regione attualmente visualizzata. Quindi, scegli la regione a cui vuoi passare.
3. Nel pannello di navigazione a sinistra, scegli gruppo Storage Lens.
4. Scegli Crea gruppo Storage Lens.
5. In Generale, visualizza la tua regione di residenza e inserisci il nome del gruppo Storage Lens.
6. In Ambito, scegli il filtro da applicare al gruppo Storage Lens. Per applicare più filtri, seleziona i filtri, quindi scegli l'operatore logico AND oppure OR.
   - Per il filtro Prefissi, scegli Prefissi e inserisci una stringa di prefisso. Per aggiungere più prefissi, scegli Aggiungi prefisso. Per rimuovere un prefisso, scegli Rimuovi accanto al prefisso che desideri eliminare.
   - Per il filtro Tag di oggetti, scegli Tag di oggetti e inserisci la coppia chiave-valore per l'oggetto. Quindi scegli Aggiungi tag. Per rimuovere un tag, scegli Rimuovi accanto al tag che desideri eliminare.
   - Per il filtro Suffissi, scegli Suffissi. e inserisci una stringa di suffisso. Per aggiungere più suffissi, scegli Aggiungi suffisso. Per rimuovere un suffisso, scegli Rimuovi accanto al suffisso che desideri eliminare.
   - Per il filtro Età, specifica l'intervallo di età dell'oggetto in giorni. Scegli Specifica l'età minima dell'oggetto e inserisci l'età minima dell'oggetto. Poi scegli Specifica l'età massima dell'oggetto e inserisci l'età massima dell'oggetto.
   - Per il filtro Dimensione, specifica l'intervallo di dimensioni dell'oggetto e l'unità di misura. Scegli Specifica la dimensione minima dell'oggetto e inserisci la dimensione minima dell'oggetto. Poi scegli Specifica la dimensione massima dell'oggetto e inserisci la dimensione dell'oggetto.
7. (Facoltativo) Per i tag AWS delle risorse, aggiungi la coppia chiave-valore, quindi scegli Aggiungi tag.
8. Scegli Crea gruppo Storage Lens.

Usando il AWS CLI

Il AWS CLI comando di esempio seguente crea un gruppo Storage Lens. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
aws s3control create-storage-lens-group --account-id 111122223333 \
--region us-east-1 --storage-lens-group=file://./marketing-department.json
```

Il AWS CLI comando di esempio seguente crea un gruppo Storage Lens con due tag di AWS risorsa. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
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
```

Per esempi di configurazione JSON, consulta Configurazione dei gruppi Storage Lens.

Utilizzo dell' AWS SDK for Java

L' AWS SDK for Java esempio seguente crea un gruppo Storage Lens. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

Example – Creare un gruppo Storage Lens con un solo filtro

Nel seguente esempio viene creato un gruppo Storage Lens denominato *Marketing-Department*. Questo gruppo dispone di un filtro relativo all'età degli oggetti che specifica una fascia di età compresa tra *30* e *90* giorni. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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 – Creare un gruppo Storage Lens con un operatore **AND** che include più filtri

Nel seguente esempio viene creato un gruppo Storage Lens denominato *Marketing-Department*. Per questo gruppo viene utilizzato l'operatore AND per indicare che gli oggetti devono soddisfare tutte le condizioni del filtro. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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 – Creare un gruppo Storage Lens con un operatore **OR** che include più filtri

Nel seguente esempio viene creato un gruppo Storage Lens denominato *Marketing-Department*. Per questo gruppo viene utilizzato un operatore OR per applicare un filtro di prefisso (*prefix-1*, *prefix-2*, *prefix3/sub-prefix-1*) o un filtro per le dimensioni degli oggetti con un intervallo di dimensioni compreso tra *1000* e *6000* byte. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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 — Crea un gruppo Storage Lens con un solo filtro e due tag di AWS risorsa

Nel seguente esempio viene creato un gruppo Storage Lens denominato *Marketing-Department* e dotato di un filtro di suffisso. Questo esempio aggiunge anche due tag di AWS risorsa al gruppo Storage Lens. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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();
```

```
        }
    }
}
```

Per esempi di configurazione JSON, consulta Configurazione dei gruppi Storage Lens.

Collegare o rimuovere gruppi S3 Storage Lens a o da un pannello di controllo

Dopo aver effettuato l'upgrade al livello avanzato di Amazon S3 Storage Lens, puoi collegare un gruppo Storage Lens al pannello di controllo. Se hai diversi gruppi Storage Lens, puoi includere o escludere i gruppi desiderati.

I gruppi Storage Lens devono risiedere nella regione di origine designata nell'account del pannello di controllo. Dopo aver collegato un gruppo Storage Lens al pannello di controllo, riceverai i dati aggiuntivi relativi all'aggregazione del gruppo Storage Lens nel documento di esportazione dei parametri entro 48 ore.

> **Note**
>
> Se desideri vedere le metriche aggregate per il gruppo Storage Lens, devi collegare quest'ultimo al tuo pannello di controllo Storage Lens. Per esempi di file di configurazione JSON del gruppo Storage Lens, consulta Esempio di configurazione di S3 Storage Lens con gruppi Storage Lens in JSON.

Collegare un gruppo Storage Lens a un pannello di controllo S3 Storage Lens

Per collegare un gruppo Storage Lens a un pannello di controllo Storage Lens

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, in Storage Lens scegli Pannelli di controllo.
3. Scegli il pulsante di opzione per il pannello di controllo Storage Lens a cui desideri collegare un gruppo Storage Lens.
4. Scegliere Modifica.
5. Sotto Metrics selection (Selezione dei parametri), scegli Advanced metrics and recommendations (Raccomandazioni e parametri avanzati).
6. Seleziona Aggregazione del gruppo Storage Lens.

> **Note**
>
> I parametri avanzati sono selezionati per impostazione predefinita. Tuttavia, è anche possibile deselezionare questa impostazione, poiché non è necessaria per aggregare i dati dei gruppi di Storage Lens.

7. Scorri verso il basso fino ad Aggregazione del gruppo Storage Lens e specifica il gruppo o i gruppi Storage Lens che desideri includere o escludere nell'aggregazione dei dati. È possibile utilizzare una qualsiasi delle seguenti opzioni di filtro:
   - Se desideri includere determinati gruppi di Storage Lens, scegli Includi gruppi Storage Lens. In Gruppi Storage Lens da includere, seleziona i tuoi gruppi Storage Lens.
   - Se desideri includere tutti i gruppi Storage Lens, seleziona Includi tutti i gruppi Storage Lens nella regione di origine di questo account.
   - Se desideri escludere determinati gruppi Storage Lens, scegli Escludi gruppi Storage Lens. In Gruppi Storage Lens da escludere, seleziona i gruppi Storage Lens che desideri escludere.
8. Seleziona Salva modifiche. Se hai configurato correttamente i gruppi Storage Lens, vedrai i dati di aggregazione aggiuntivi del gruppo Storage Lens nel tuo pannello di controllo entro 48 ore.

Rimuovere un gruppo Storage Lens da un pannello di controllo S3 Storage Lens

Per rimuovere un gruppo Storage Lens da un pannello di controllo S3 Storage Lens

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, in Storage Lens scegli Pannelli di controllo.
3. Scegli il pulsante di opzione per il pannello di controllo Storage Lens dal quale desideri rimuovere un gruppo Storage Lens.
4. Scegli Visualizza configurazione del pannello di controllo.
5. Scegliere Modifica.
6. Scorri verso il basso fino alla sezione Selezione di parametri.
7. In Aggregazione del gruppo Storage Lens, scegli la X accanto al gruppo Storage Lens che desideri rimuovere. Al termine di questa operazione il gruppo Storage Lens viene rimosso.

Se nel pannello di controllo hai incluso tutti i gruppi Storage Lens, deseleziona la casella accanto a Includi tutti i gruppi Storage Lens nella regione di origine di questo account.

8. Seleziona Salva modifiche.

> **Note**
>
> Sono necessarie fino a 48 ore prima che il pannello di controllo rifletta gli aggiornamenti di configurazione.

Visualizzazione dei dati dei gruppi Storage Lens

Puoi visualizzare i dati dei gruppi Storage Lens collegando il gruppo al pannello di controllo di Amazon S3 Storage Lens. Dopo aver incluso il gruppo Storage Lens nell'aggregazione del gruppo Storage Lens nella configurazione del pannello di controllo, possono essere necessarie fino a 48 ore prima che i dati del gruppo Storage Lens vengano visualizzati nel pannello di controllo.

Una volta aggiornata la configurazione del pannello di controllo, tutti i gruppi Storage Lens appena collegati vengono visualizzati nell'elenco delle risorse disponibili nella scheda Gruppi Storage Lens. È inoltre possibile analizzare ulteriormente l'utilizzo dell'archiviazione nella scheda Panoramica suddividendo in base ad altre dimensioni. Ad esempio, puoi scegliere uno degli elementi elencati nelle prime 3 categorie e scegliere Analizza per per suddividere i dati in base a un'altra dimensione. Non è possibile applicare la stessa dimensione del filtro stesso.

> **Note**
>
> Non è possibile applicare un filtro del gruppo Storage Lens insieme a un filtro con prefisso o viceversa. Inoltre, non è possibile analizzare ulteriormente un gruppo Storage Lens utilizzando un filtro con prefisso.

Puoi utilizzare la scheda Gruppo Storage Lens nel pannello di controllo di Amazon S3 Storage Lens per personalizzare la visualizzazione dei dati per i gruppi di Storage Lens collegati al pannello di controllo. Puoi visualizzare i dati di tutti gruppi Storage Lens collegati al pannello di controllo o solo di alcuni.

Quando visualizzi i dati del gruppo Storage Lens nel pannello di controllo di S3 Storage Lens, tieni presente quanto segue:

- S3 Storage Lens aggrega i parametri di utilizzo di un oggetto in tutti i gruppi Storage Lens corrispondenti. Pertanto, se un oggetto soddisfa le condizioni di filtro per due o più gruppi Storage Lens, i conteggi relativi all'utilizzo dell'archiviazione saranno visualizzati come ripetuti per lo stesso oggetto.
- Gli oggetti devono corrispondere ai filtri che includi nei gruppi Storage Lens. Se nessun oggetto corrisponde ai filtri che includi nel gruppo Storage Lens, non verrà generato alcun parametro. Per determinare se ci sono oggetti non assegnati, controlla il numero totale di oggetti nel pannello di controllo a livello di account e di bucket.

Aggiornamento di un gruppo Storage Lens

I seguenti esempi illustrano come aggiornare un gruppo Amazon S3 Storage Lens. Puoi aggiornare un gruppo Storage Lens utilizzando la console Amazon S3, la AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

Per aggiornare un gruppo Storage Lens

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Gruppi Storage Lens.
3. In Gruppi Storage Lens, scegli il gruppo Storage Lens che desideri aggiornare.
4. In Ambito, scegli Modifica.
5. Nella pagina Ambito, scegli il filtro da applicare al gruppo Storage Lens. Per applicare più filtri, seleziona i filtri, quindi scegli l'operatore logico AND oppure OR.
    - Per il filtro Prefissi, seleziona Prefissi e inserisci una stringa di prefisso. Per aggiungere più prefissi, scegli Aggiungi prefisso. Per rimuovere un prefisso, scegli Rimuovi accanto al prefisso che desideri eliminare.
    - Per il filtro Tag di oggetti, inserisci la coppia chiave-valore per l'oggetto. Quindi scegli Aggiungi tag. Per rimuovere un tag, scegli Rimuovi accanto al tag che desideri eliminare.
    - Per il filtro Suffissi, seleziona Suffissi. e inserisci una stringa di suffisso. Per aggiungere più suffissi, scegli Aggiungi suffisso. Per rimuovere un suffisso, scegli Rimuovi accanto al suffisso che desideri eliminare.

- Per il filtro Età, specifica l'intervallo di età dell'oggetto in giorni. Scegli Specifica l'età minima dell'oggetto e inserisci l'età minima dell'oggetto. In Specifica l'età massima dell'oggetto inserisci l'età massima dell'oggetto.
- Per il filtro Dimensione, specifica l'intervallo di dimensioni dell'oggetto e l'unità di misura. Scegli Specifica la dimensione minima dell'oggetto e inserisci la dimensione minima dell'oggetto. In Specifica la dimensione massima dell'oggetto inserisci la dimensione dell'oggetto.

6. Seleziona Salva modifiche. Viene visualizzata la pagina dei dettagli del gruppo Storage Lens.
7. (Facoltativo) Se desideri aggiungere un nuovo tag delle risorse AWS, scorri fino alla sezione dei Tag delle risorse AWS, quindi scegli Aggiungi tag. Viene visualizzata la pagina Aggiungi tag.

   Aggiungi la nuova coppia chiave-valore, quindi scegli Salva modifiche. Viene visualizzata la pagina dei dettagli del gruppo Storage Lens.
8. (Facoltativo) Se desideri rimuovere un tag delle risorse AWS esistente, scorri fino alla sezione dei Tag delle risorse AWS e seleziona il tag della risorsa. Quindi, scegli Elimina. Viene visualizzata la finestra di dialogo Elimina i tag AWS.

   Scegli nuovamente Elimina per eliminare definitivamente il tag delle risorse AWS.

   > **Note**
   >
   > Una volta che un tag delle risorse AWS è stato eliminato in modo permanente, non è possibile ripristinarlo.

Utilizzo di AWS CLI

Il comando AWS CLI di esempio seguente restituisce i dettagli di configurazione per un gruppo Storage Lens denominato *marketing-department*. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
aws s3control get-storage-lens-group --account-id 111122223333 \
--region us-east-1 --name marketing-department
```

Il seguente comando AWS CLI di esempio viene utilizzato per aggiornare un gruppo Storage Lens. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
aws s3control update-storage-lens-group --account-id 111122223333 \
--region us-east-1 --storage-lens-group=file://./marketing-department.json
```

Per esempi di configurazione JSON, consulta Configurazione dei gruppi Storage Lens.

Utilizzo dell'SDK AWS per Java

Il comando AWS SDK for Java di esempio seguente restituisce i dettagli di configurazione per il gruppo Storage Lens *Marketing-Department* nell'account *111122223333*. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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();
        }
    }
}
```

Nell'esempio seguente viene aggiornato il gruppo Storage Lens *Marketing-Department* nell'account *111122223333*. In questo esempio l'ambito del pannello di controllo viene aggiornato per includere oggetti che corrispondano a uno dei seguenti suffissi: *.png*, *.gif*, *.jpg* o *.jpeg*. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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();
        }
    }
}
```

Per esempi di configurazione JSON, consulta Configurazione dei gruppi Storage Lens.

Gestione dei tag delle risorse AWS con i gruppi Storage Lens

Ogni gruppo Amazon S3 Storage Lens viene conteggiato come una risorsa AWS con un proprio nome della risorsa Amazon (ARN). Pertanto, quando configuri il gruppo Storage Lens, puoi aggiungere facoltativamente tag delle risorse AWS al gruppo. È possibile aggiungere fino a 50 tag per ogni gruppo Storage Lens. Per creare un gruppo Storage Lens con tag, devi disporre delle autorizzazioni `s3:CreateStorageLensGroup` e `s3:TagResource`.

È possibile utilizzare i tag delle risorse AWS per classificare le risorse in base al reparto, alla linea di business o al progetto. Ciò è utile quando si dispone di numerose risorse dello stesso tipo. Applicando i tag, è possibile individuare rapidamente un gruppo Storage Lens in base ai tag che hai assegnato loro. È possibile utilizzare i tag anche per monitorare e allocare i costi.

Inoltre, quando aggiungi un tag delle risorse AWS al tuo gruppo Storage Lens, attivi il controllo degli accessi basato su attributi (ABAC). L'ABAC è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi, in questo caso ai tag. È possibile utilizzare le condizioni per specificare i tag delle risorse nelle policy IAM per controllare l'accesso alle risorse AWS.

Puoi modificare chiavi e valori di tag e rimuovere tag da una risorsa in qualsiasi momento. Inoltre, tieni presente le limitazioni seguenti:

- I valori e le chiavi dei tag rispettano la distinzione tra maiuscole e minuscole.
- Se aggiungi un tag con la stessa chiave di un tag esistente a una risorsa specifica, il nuovo valore sovrascrive quello precedente.
- Se elimini una risorsa, verranno eliminati anche tutti i tag associati alla risorsa.
- Non includere dati privati o sensibili nei tag delle risorse AWS.
- I tag di sistema (con chiavi di tag che iniziano con `aws:`) non sono supportati.
- La lunghezza di ogni chiave di tag non può superare i 128 caratteri. La lunghezza di ogni valore di tag non può superare i 256 caratteri.

I seguenti esempi illustrano come utilizzare i tag delle risorse AWS con i gruppi Storage Lens.

Argomenti

Aggiungere un tag delle risorse AWS a un gruppo Storage Lens

I seguenti esempi illustrano come aggiungere tag delle risorse AWS a un gruppo Amazon S3 Storage Lens. Puoi aggiungere tag di risorsa utilizzando la console Amazon S3, la AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

Per aggiungere un tag delle risorse AWS a un gruppo Storage Lens

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Gruppi Storage Lens.
3. In Gruppi Storage Lens, scegli il gruppo Storage Lens che desideri aggiornare.
4. In Tag delle risorse AWS seleziona Aggiungi tag.
5. Nella pagina Aggiungi tag, aggiungi la nuova coppia chiave-valore.

> **Note**
>
> Aggiungendo un tag la cui chiave è la stessa di un tag esistente viene sovrascritto il valore del tag precedente.

6. (Facoltativo) Per aggiungere più di un nuovo tag, scegliete nuovamente Aggiungi tag e aggiungi nuove voci. È possibile aggiungere fino a 50 tag delle risorse AWS al gruppo Storage Lens.
7. (Facoltativo) Se desideri rimuovere un tag appena aggiunto, scegli Rimuovi accanto al tag che desideri eliminare.
8. Seleziona Salva modifiche.

Utilizzo di AWS CLI

Con il comando AWS CLI di esempio seguente vengono aggiunti due tag di risorsa a un gruppo Storage Lens esistente denominato *marketing-department*. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
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
```

Utilizzo dell'SDK AWS per Java

Con il comando AWS SDK for Java di esempio seguente vengono aggiunti due tag delle risorse AWS a un gruppo Storage Lens esistente. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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();
        }
    }
}
```

Aggiornamento dei valori dei tag di un gruppo Storage Lens

Gli esempi seguenti mostrano come aggiornare i valori di tag di un gruppo Storage Lens utilizzando la console Amazon S3, la AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

Per aggiungere un tag delle risorse AWS a un gruppo Storage Lens

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Gruppi Storage Lens.
3. In Gruppi Storage Lens, scegli il gruppo Storage Lens che desideri aggiornare.
4. In Tag delle risorse AWS, seleziona il tag che desideri aggiornare.
5. Aggiungi il nuovo valore del tag, utilizzando la stessa chiave della coppia chiave-valore che desideri aggiornare. Seleziona l'icona del segno di spunta per aggiornare il valore del tag.

   > **Note**
   >
   > Aggiungendo un tag la cui chiave è la stessa di un tag esistente viene sovrascritto il valore del tag precedente.

6. (Facoltativo) Se desideri aggiungere nuovi tag, scegli Aggiungi tag per aggiungere nuove voci. Viene visualizzata la pagina Add tags (Aggiungi tag).

   È possibile aggiungere fino a 50 tag delle risorse AWS al gruppo Storage Lens. Una volta finito di aggiungere i tag, scegli Salva modifiche.
7. (Facoltativo) Se desideri rimuovere un tag appena aggiunto, scegli Rimuovi accanto al tag che desideri eliminare. Una volta finito di rimuovere i tag, scegli Salva modifiche.

Utilizzo di AWS CLI

Con il comando AWS CLI di esempio seguente vengono aggiornati due valori di tag per il gruppo Storage Lens denominato *`marketing-department`*. Per utilizzare questo comando di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

```
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
```

Utilizzo dell'SDK AWS per Java

Con l'esempio AWS SDK for Java seguente vengono aggiornati due valori di tag del gruppo Storage Lens. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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();
        }
    }
}
```

Eliminazione di un tag delle risorse AWS da un gruppo Storage Lens

I seguenti esempi illustrano come eliminare tag delle risorse AWS da un gruppo Storage Lens. Puoi eliminare tag di risorsa utilizzando la console Amazon S3, la AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

Per eliminare un tag delle risorse AWS da un gruppo Storage Lens

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Gruppi Storage Lens.
3. In Gruppi Storage Lens, scegli il gruppo Storage Lens che desideri aggiornare.
4. In Tag delle risorse AWS, seleziona la coppia chiave-valore che desideri eliminare.
5. Scegli Elimina. Viene visualizzata la finestra di dialogo Elimina i tag delle risorse AWS.

> **Note**
>
> Se si utilizzano tag per controllare l'accesso, effettuare questa operazione può influire sulle risorse correlate. Una volta eliminato definitivamente, non è possibile ripristinare un tag.

6. Scegli Elimina per eliminare la coppia chiave-valore in modo permanente.

Utilizzo di AWS CLI

Con il comando AWS CLI seguente vengono eliminati due tag delle risorse AWS da un gruppo Storage Lens esistente. Per utilizzare questo comando di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

```
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
```

Utilizzo dell'SDK AWS per Java

Nell'esempio AWS SDK for Java seguente vengono eliminati due tag delle risorse AWS dal nome della risorsa Amazon (ARN) del gruppo Storage Lens specificato nell'account *111122223333*. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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();
        }
    }
}
```

Elenco dei tag dei gruppi Storage Lens

I seguenti esempi illustrano come elencare i tag delle risorse AWS associate a un gruppo Storage Lens. Puoi elencare tag di risorsa utilizzando la console Amazon S3, la AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

Per esaminare l'elenco e i valori dei tag per un gruppo Storage Lens

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Gruppi Storage Lens.
3. In Gruppi Storage Lens, scegli il gruppo Storage Lens di tuo interesse.
4. Scorri verso il basso fino alla sezione Tag delle risorse AWS. Tutti i tag delle risorse AWS definiti dall'utente aggiunti al gruppo Storage Lens sono elencati insieme ai relativi valori di tag.

Utilizzo di AWS CLI

Con il comando AWS CLI di esempio seguente vengono elencati tutti i valori di tag per il gruppo Storage Lens denominato *marketing-department*. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
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
```

Utilizzo dell'SDK AWS per Java

Nell'esempio AWS SDK for Java seguente vengono elencati i valori dei tag del gruppo Storage Lens per il nome della risorsa Amazon (ARN) del gruppo Storage Lens specificato. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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();
        }
    }
}
```

Elenco di tutti i gruppi Storage Lens

Negli esempi seguenti viene illustrato come elencare tutti i gruppi Amazon S3 Storage Lens in un Account AWS e una regione di origine. Questi esempi mostrano come elencare tutti i gruppi di Storage Lens utilizzando la console Amazon S3, la AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

Per elencare tutti i gruppi Storage Lens in un account e in una regione di origine

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Gruppi Storage Lens.
3. In Gruppi Storage Lens, viene visualizzato l'elenco dei gruppi Storage Lens presenti nell'account.

Utilizzo di AWS CLI

Nell'esempio AWS CLI seguente vengono elencati tutti i gruppi Storage Lens del tuo account. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
aws s3control list-storage-lens-groups --account-id 111122223333 \
--region us-east-1
```

Utilizzo dell'SDK AWS per Java

Nell'esempio AWS SDK for Java seguente vengono elencati tutti i gruppi Storage Lens dell'account *111122223333*. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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();
        }
    }
}
```

Visualizzazione dei dettagli del gruppo Storage Lens

I seguenti esempi illustrano come visualizzare i dettagli di un gruppo Amazon S3 Storage Lens. Puoi visualizzare tali dettagli utilizzando la console Amazon S3, la AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

Per visualizzare i dettagli di configurazione del gruppo Storage Lens

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

2. Nel pannello di navigazione a sinistra, scegli Gruppi Storage Lens.
3. In Gruppi Storage Lens, scegli il pulsante di opzione accanto al gruppo Storage Lens di tuo interesse.
4. Seleziona Visualizza dettagli. Ora puoi rivedere i dettagli del tuo gruppo Storage Lens.

Utilizzo di AWS CLI

Nell'esempio AWS CLI seguente vengono restituiti i dettagli di configurazione per un gruppo Storage Lens. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
aws s3control get-storage-lens-group --account-id 111122223333 \
--region us-east-1 --name marketing-department
```

Utilizzo dell'SDK AWS per Java

Il comando AWS SDK for Java di esempio seguente restituisce i dettagli di configurazione per il gruppo Storage Lens *Marketing-Department* nell'account *111122223333*. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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();
        }
    }
}
```

Eliminazione di un gruppo Storage Lens

Gli esempi seguenti mostrano come eliminare un gruppo Amazon S3 Storage Lens utilizzando la console Amazon S3, la AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

Per eliminare un gruppo Storage Lens

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Gruppi Storage Lens.
3. In Gruppi Storage Lens, scegli il pulsante di opzione accanto al gruppo Storage Lens che desideri eliminare.
4. Scegli Elimina. Viene visualizzata la finestra di dialogo Elimina gruppo Storage Lens.
5. Scegli nuovamente Elimina per rimuovere definitivamente il gruppo Storage Lens.

> **Note**
>
> Uno volta eliminato, non è più possibile ripristinare un gruppo Storage Lens.

Utilizzo di AWS CLI

Nell'esempio AWS CLI seguente viene eliminato il gruppo Storage Lens denominato *marketing-department*. Per utilizzare questo comando di esempio, sostituisci *user input placeholders* con le tue informazioni.

```
aws s3control delete-storage-lens-group --account-id 111122223333 \
--region us-east-1 --name marketing-department
```

Utilizzo dell'SDK AWS per Java

Nell'esempio AWS SDK for Java seguente viene eliminato il gruppo Storage Lens denominato *Marketing-Department* nell'account *111122223333*. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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();
        }
    }
}
```

Tracciamento delle richieste Amazon S3 tramite AWS X-Ray

AWS X-Ray raccoglie i dati relativi alle richieste dall'applicazione. Puoi quindi visualizzare e filtrare i dati per identificare e risolvere i problemi di prestazioni e gli errori nelle applicazioni distribuite e nell'architettura dei microservizi. Per qualsiasi richiesta tracciata che raggiunge la tua applicazione, puoi visualizzare informazioni dettagliate sulla richiesta, sulla risposta e sulle chiamate che la tua applicazione esegue verso le risorse AWS, i microservizi, i database e le API Web HTTP downstream.

Per ulteriori informazioni, consulta Che cos'è AWS X-Ray nella Guida per sviluppatori di AWS X-Ray.

Argomenti

- Come funziona X-Ray con Amazon S3
- Regioni disponibili

Come funziona X-Ray con Amazon S3

AWS X-Ray supporta la propagazione del contesto di traccia per Amazon S3, in modo da poter visualizzare le richieste end-to-end mentre si muovono all'interno dell'applicazione. X-Ray aggrega i dati generati dai servizi individuali come Amazon S3, AWS Lambda e Amazon EC2 e le molte risorse che compongono la tua applicazione. Fornisce una visione generale delle prestazioni dell'applicazione.

Amazon S3 si integra con X-Ray per propagare il contesto di traccia e fornire una catena di richieste con nodi upstream e downstream . Se un servizio upstream con la sua richiesta S3 include un'intestazione di traccia in formato valido, Amazon S3 passa l'intestazione di traccia quando invia notifiche di eventi ai servizi downstream come Lambda, Amazon SQS e Amazon SNS. Se hai tutti questi servizi attivamente integrati con X-Ray, questi sono collegati in un'unica catena di richieste per fornirti i dettagli completi delle tue richieste Amazon S3.

Per inviare intestazioni di traccia X-Ray tramite Amazon S3, è necessario includere un X-Amzn-Trace-Id formattato nelle richieste. Puoi anche strumentare il client Amazon S3 utilizzando gli SDK AWS X-Ray. Per un elenco degli SDK supportati, consulta la Documentazione su AWS X-Ray.

Mappe dei servizi

Le mappe dei servizi X-Ray mostrano le relazioni tra Amazon S3 e altri servizi e risorse AWS nella tua applicazione in tempo quasi reale. Per visualizzare le richieste end-to-end utilizzando le mappe dei servizi X-Ray, puoi utilizzare la console X-Ray per visualizzare una mappa delle connessioni tra Amazon S3 e altri servizi utilizzati dall'applicazione. Puoi rilevare facilmente in quale punto dell'applicazione avvengono le latenze elevate, visualizzare la distribuzione per questi servizi ed eseguire il drill-down in servizi e percorsi specifici che influenzano le prestazioni dell'applicazione.

Analisi X-Ray

Puoi utilizzare la console Analisi X-Ray per analizzare le tracce, visualizzare parametri quali latenza e percentuale di errori e generare informazioni utili per identificare e risolvere i problemi. Questa console riporta inoltre parametri quali la latenza media e le percentuali di errore. Per ulteriori informazioni, consulta la sezione Console AWS X-Ray nella Guida per gli sviluppatori AWS X-Ray.

Regioni disponibili

Il supporto di AWS X-Ray per Amazon S3 è disponibile in tutte le Regioni AWS X-Ray. Per ulteriori informazioni, consulta la sezione Amazon S3 e AWS X-Ray nella Guida per gli sviluppatori AWS X-Ray.

Hosting di un sito Web statico tramite Amazon S3

Puoi utilizzare Amazon S3 per ospitare un sito web statico. In un sito Web statico, le singole pagine Web includono contenuti statici. Potrebbero contenere anche script lato client.

Un sito Web dinamico, al contrario, si basa sull'elaborazione lato server, inclusi gli script lato server come PHP, JSP o ASP.NET. Amazon S3 non supporta lo scripting lato server, ma AWS dispone di altre risorse per l'hosting di siti Web dinamici. Per ulteriori informazioni sull'hosting di siti Web su AWS, consulta Web Hosting.

> **Note**
>
> Puoi utilizzare la AWS Amplify Console per ospitare un'app Web a pagina singola. La console AWS Amplify supporta app a pagina singola create con framework di applicazioni a pagina singola, ad esempio, React JS, Vue JS, Angular JS e Nuxt, e generatori di siti statici, ad esempio, Gatsby JS, React-Static, Jekyll e Hugo. Per ulteriori informazioni, consulta la sezione Nozioni di base nella Guida per l'utente della console AWS Amplify .
> Gli endpoint dei siti web Amazon S3 non supportano HTTPS. Se desideri utilizzare HTTPS, puoi utilizzare Amazon CloudFront per servire un sito Web statico ospitato su Amazon S3. Per ulteriori informazioni, consulta Come posso utilizzare CloudFront per servire le richieste HTTPS per il mio bucket Amazon S3? Per utilizzare HTTPS con un dominio personalizzato, consulta Configurazione di un sito Web statico utilizzando un dominio personalizzato registrato con Route 53.

Per ulteriori informazioni sull'hosting di un sito Web statico su Amazon S3, incluse istruzioni e procedure dettagliate, step-by-step consulta i seguenti argomenti.

Argomenti

Endpoint del sito Web

Quando configuri il bucket come sito Web statico, il sito Web è disponibile nell'endpoint del sito Web specifico della Regione AWS del bucket. Gli endpoint dei siti Web sono diversi dagli endpoint dove si inviano le richieste REST API. Per ulteriori informazioni sulle differenze tra gli endpoint, consulta Differenze chiave tra un endpoint del sito Web e un endpoint REST API.

A seconda della regione, gli endpoint del sito web Amazon S3 seguono uno di questi due formati.

- Regione s3-website dash - `http://bucket-name.s3-website-Region.amazonaws.com`
- s3-website dot (.) Regione - `http://bucket-name.s3-website.Region.amazonaws.com`

Questi URL restituiscono un documento di indice predefinito che si configura per il sito Web. Per un elenco completo degli endpoint dei siti Web Amazon S3, consulta la sezione Endpoint di siti Web Amazon S3.

> **Note**
>
> Per aumentare la sicurezza dei siti Web statici di Amazon S3, i domini endpoint dei siti Web Amazon S3 (ad esempio, s3 website-us-east - -1.amazonaws.com o s3-website.ap-south-1.amazonaws.com) sono registrati nella Public Suffix List (PSL). Per una maggiore sicurezza, consigliamo di utilizzare i cookie con un prefisso `__Host-` se hai bisogno di impostare cookie sensibili nel nome di dominio per i siti Web statici Amazon S3. Questa pratica ti aiuterà a difendere il tuo dominio dai tentativi CSRF (cross-site request forgery). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

Se desideri che il sito Web sia pubblico, è necessario rendere tutti i contenuti pubblicamente leggibili affinché i clienti possano accedervi nell'endpoint del sito Web. Per ulteriori informazioni, consulta Impostazione delle autorizzazioni per l'accesso al sito Web.

> **Important**
>
> Gli endpoint del sito Web di Amazon S3 non supportano HTTPS o access point. Se desideri utilizzare HTTPS, puoi utilizzare Amazon CloudFront per servire un sito Web statico ospitato su Amazon S3. Per ulteriori informazioni, consulta Come posso utilizzare CloudFront per servire le richieste HTTPS per il mio bucket Amazon S3? Per utilizzare HTTPS con un

dominio personalizzato, consulta Configurazione di un sito Web statico utilizzando un dominio personalizzato registrato con Route 53.
I bucket con pagamento a carico del richiedente non consentono l'accesso tramite un endpoint di sito Web. Qualsiasi richiesta a tale bucket riceve una risposta 403 Accesso negato . Per ulteriori informazioni, consulta Utilizzo dei bucket con pagamento a carico del richiedente per utilizzo e trasferimenti di storage.

Argomenti

Esempi di endpoint del sito Web

Negli esempi seguenti viene illustrato come è possibile accedere a un bucket Amazon S3 configurato come sito web statico.

Example – Richiesta di un oggetto a livello root

Per richiedere un oggetto specifico archiviato a livello root nel bucket, utilizza la seguente struttura di URL:

```
http://bucket-name.s3-website.Region.amazonaws.com/object-name
```

Ad esempio, questo URL richiede l'oggetto `photo.jpg` archiviato a livello root nel bucket:

```
http://example-bucket.s3-website.us-west-2.amazonaws.com/photo.jpg
```

Example – Richiesta di un oggetto in un prefisso

Per richiedere un oggetto archiviato in una cartella nel bucket, utilizza questa struttura di URL:

```
http://bucket-name.s3-website.Region.amazonaws.com/folder-name/object-name
```

Il seguente URL richiede l'oggetto `docs/doc1.html` nel bucket.

```
http://example-bucket.s3-website.us-west-2.amazonaws.com/docs/doc1.html
```

Aggiunta di un CNAME DNS

Se si dispone di un dominio registrato, è possibile aggiungere una voce DNS CNAME che punti all'endpoint del sito web Amazon S3. Ad esempio, se hai registrato il dominio `www.example-bucket.com`, puoi creare un bucket `www.example-bucket.com` e aggiungere un record DNS CNAME che punti a `www.example-bucket.com.s3-website.Region.amazonaws.com`. Tutte le richieste a `http://www.example-bucket.com` vengono instradate verso `www.example-bucket.com.s3-website.Region.amazonaws.com`.

Per ulteriori informazioni, consulta Personalizzazione degli URL Amazon S3 con record CNAME.

Utilizzo di un dominio personalizzato con Route 53

Invece di accedere al sito web utilizzando un endpoint del sito web Amazon S3, è possibile utilizzare il proprio dominio registrato con Amazon Route 53 per servire i contenuti, ad esempio, `example.com`. Puoi utilizzare Amazon S3 con Route 53 per ospitare un sito web nel dominio principale. Ad esempio, se si dispone di un dominio root `example.com` e si ospita il sito web su Amazon S3, i visitatori del sito web possono accedere al sito dal loro browser, inserendo `http://www.example.com` o `http://example.com`.

Per un esempio di procedura guidata, consulta Tutorial: Configurazione di un sito Web statico utilizzando un dominio personalizzato registrato con Route 53.

Differenze chiave tra un endpoint del sito Web e un endpoint REST API

L'endpoint del sito web Amazon S3 è ottimizzato per l'accesso da un browser web. Nella tabella seguente vengono riepilogate le principali differenze tra un endpoint REST API e un endpoint del sito Web.

Differenze principali	Endpoint REST API	Endpoint del sito Web
Controllo degli accessi	Supporta contenuti pubblici e privati.	Supporta solo contenuti pubblicam ente leggibili.

Differenze principali	Endpoint REST API	Endpoint del sito Web
Gestione dei messaggi di errore	Restituisce una risposta di errore in formato XML.	Restituisce un documento HTML.
Supporto del reindirizzamento	Non applicabile.	Supporta reindirizzamenti sia a livello di oggetto sia di bucket.
Richieste supportate	Supporta tutte le operazioni relative ai bucket e agli oggetti.	Supporta solo richieste GET e HEAD su oggetti.
Risposte alle richieste GET e HEAD alla root di un bucket	Restituisce un elenco delle chiavi degli oggetti nel bucket.	Restituisce un documento di indice specificato nella configurazione del sito Web.
Supporto di Secure Sockets Layer (SSL)	Supporta connessioni SSL.	Non supporta connessioni SSL.

Per un elenco completo degli endpoint Amazon S3, consultare la sezione relativa a endpoint e quote di Amazon S3 nella Riferimenti generali di AWS.

Abilitazione dell'hosting di siti Web

Quando configuri un bucket come sito Web statico, devi abilitare l'hosting statico del sito Web, configurare un documento di indice e impostare le autorizzazioni.

Puoi abilitare l'hosting di siti Web statici utilizzando la console Amazon S3, l'API REST, gli AWS SDK, o. AWS CLI AWS CloudFormation

Per configurare il sito Web con un dominio personalizzato, consulta Tutorial: Configurazione di un sito Web statico utilizzando un dominio personalizzato registrato con Route 53.

Utilizzo della console S3

Per abilitare l'hosting di un sito Web statico

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Nome bucket, seleziona il nome del bucket per cui desideri abilitare l'hosting di siti Web statici.
3. Scegliere Properties (Proprietà).
4. In Hosting di siti Web statici, seleziona Modifica.
5. Seleziona Utilizza questo bucket per l'hosting di un sito Web.
6. In Hosting di siti Web statici, seleziona Abilita.
7. In Documento di indice immettere il nome file del documento di indice, in genere `index.html`.

   Il nome del documento indice fa distinzione tra maiuscole e minuscole e deve corrispondere esattamente al nome del file del documento indice HTML che si prevede di caricare nel bucket S3. Quando si configura un bucket per l'hosting di siti Web, è necessario specificare un documento di indice. Amazon S3 restituisce questo documento di indice quando si eseguono richieste per il dominio root o per una delle sottocartelle. Per ulteriori informazioni, consulta Configurazione di un documento indice.
8. Per fornire il tuo documento di errore personalizzato per gli errori di classe 4XX, specifica il nome file del documento in Documento di errore.

   Il nome del documento di errore fa distinzione tra maiuscole e minuscole e deve corrispondere esattamente al nome del file del documento di errore HTML che si prevede di caricare nel bucket S3. Se non si specifica un documento di errore personalizzato e si verifica un errore, Amazon S3 restituisce un documento di errore HTML predefinito. Per ulteriori informazioni, consulta Configurazione di un documento di errore personalizzato.
9. (Facoltativo) Per specificare regole di reindirizzamento avanzate, utilizza JSON per descrivere le regole in Regole reindirizzamento.

   Ad esempio, è possibile instradare le richieste in base a prefissi o nomi della chiave dell'oggetto specifici nella richiesta. Per ulteriori informazioni, consulta Configurazione delle regole di reindirizzamento per utilizzare i reindirizzamenti condizionali avanzati.
10. Seleziona Salva modifiche.

Amazon S3 abilita l'hosting statico del sito web per il tuo bucket. Nella parte inferiore della pagina, in Hosting di siti Web statici, viene visualizzato l'endpoint del sito web per il bucket.

11. In Hosting sito Web statico, prendi nota dell'endpoint.

    Endpoint è l'endpoint del sito web Amazon S3 per il bucket. Dopo aver configurato il bucket come sito Web statico, è possibile utilizzare questo endpoint per testare il sito Web.

Utilizzo di REST API

Per maggiori informazioni sull'invio diretto di richieste REST per abilitare l'hosting statico di siti Web, consulta le seguenti sezioni nella Guida di riferimento all'API di Amazon Simple Storage Service:

- PUT Bucket website
- GET Bucket website
- DELETE Bucket website

Utilizzo degli SDK AWS

Per ospitare un sito web statico su Amazon S3, si configura un bucket Amazon S3 per l'hosting di siti Web e, successivamente, si caricano i contenuti del sito Web nel bucket. È inoltre possibile utilizzare gli SDK AWS per creare, aggiornare ed eliminare la configurazione del sito Web a livello di codice. Gli SDK forniscono classi wrapper per REST API di Amazon S3. Se l'applicazione lo richiede, è possibile inviare richieste REST API direttamente dall'applicazione.

.NET

L'esempio seguente mostra come utilizzare per gestire la configurazione del sito Web AWS SDK for .NET per un bucket. Per aggiungere una configurazione del sito Web a un bucket, si fornisce un nome bucket e una configurazione del sito Web. La configurazione del sito Web deve includere un documento di indice e può contenere un documento di errore opzionale. Tali documenti devono essere archiviati nel bucket. Per ulteriori informazioni, consulta PUT Bucket website. Per ulteriori informazioni sulla funzionalità website di Amazon S3 consulta Hosting di un sito Web statico tramite Amazon S3.

L'esempio di codice C# seguente aggiunge una configurazione del sito Web al bucket specificato. La configurazione specifica sia il documento di indice, sia i nomi del documento di errore. Per

informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class WebsiteConfigTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string indexDocumentSuffix = "*** index object key ***"; //
 For example, index.html.
        private const string errorDocument = "*** error object key ***"; // For
 example, error.html.
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            AddWebsiteConfigurationAsync(bucketName, indexDocumentSuffix,
 errorDocument).Wait();
        }

        static async Task AddWebsiteConfigurationAsync(string bucketName,
                                            string indexDocumentSuffix,
                                            string errorDocument)
        {
            try
            {
                // 1. Put the website configuration.
                PutBucketWebsiteRequest putRequest = new PutBucketWebsiteRequest()
                {
                    BucketName = bucketName,
                    WebsiteConfiguration = new WebsiteConfiguration()
                    {
                        IndexDocumentSuffix = indexDocumentSuffix,
                        ErrorDocument = errorDocument
```

```
                    }
                };
                PutBucketWebsiteResponse response = await
 client.PutBucketWebsiteAsync(putRequest);

                // 2. Get the website configuration.
                GetBucketWebsiteRequest getRequest = new GetBucketWebsiteRequest()
                {
                    BucketName = bucketName
                };
                GetBucketWebsiteResponse getResponse = await
 client.GetBucketWebsiteAsync(getRequest);
                Console.WriteLine("Index document: {0}",
 getResponse.WebsiteConfiguration.IndexDocumentSuffix);
                Console.WriteLine("Error document: {0}",
 getResponse.WebsiteConfiguration.ErrorDocument);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

PHP

L'esempio di codice PHP seguente aggiunge una configurazione del sito Web al bucket specificato. Il metodo `create_website_config` fornisce esplicitamente il documento di indice e i nomi del documento di errore. L'esempio recupera inoltre la configurazione del sito Web e stampa la risposta. Per ulteriori informazioni sulla funzionalità website di Amazon S3 consulta Hosting di un sito Web statico tramite Amazon S3.

Per ulteriori informazioni sull'API AWS SDK for Ruby, AWS vai a SDK for Ruby - Versione 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
]);
```

Usando il AWS CLI

Per ulteriori informazioni sull'utilizzo di AWS CLI per configurare un bucket S3 come sito Web statico, consulta il sito Web nel AWS CLI Command Reference.

Successivamente, è necessario configurare il documento indice e impostare le autorizzazioni. Per informazioni, consultare Configurazione di un documento indice e Impostazione delle autorizzazioni per l'accesso al sito Web.

È inoltre possibile configurare facoltativamente un documento di errore, laregistrazione del traffico Web o un reindirizzamento.

Configurazione di un documento indice

Quando si abilita l'hosting di siti Web, è necessario configurare e caricare un documento di indice. Un documento di indice è una pagina Web che Amazon S3 restituisce quando viene fatta una richiesta alla root di un sito web o a qualsiasi sottocartella. Ad esempio, se un utente inserisce `http://www.example.com` nel browser, l'utente non richiede alcuna pagina specifica. In questo caso, Amazon S3 fornisce il documento indice, talvolta chiamato pagina predefinita.

Quando si abilita l'hosting statico di siti Web per il bucket, si immette il nome del documento di indice (ad esempi, `index.html`). Dopo aver abilitato l'hosting statico di siti Web per il bucket, si carica un file HTML con il nome del documento di indice nel bucket.

La barra finale nell'URL a livello di root è facoltativa. Ad esempio, quando si configura il sito Web con `index.html` come documento di indice, entrambi gli URL seguenti restituiscono `index.html`.

```
http://example-bucket.s3-website.Region.amazonaws.com/
http://example-bucket.s3-website.Region.amazonaws.com
```

Per ulteriori informazioni sugli endpoint del sito Amazon S3, consulta Endpoint del sito Web.

Documento di indice e cartelle

In Amazon S3, un bucket è un container flat di oggetti. Non fornisce alcuna organizzazione gerarchica in quanto è il file system del computer a farlo. Tuttavia, è possibile creare una gerarchia logica utilizzando i nomi delle chiavi degli oggetti che implicano una struttura a cartelle.

Si supponga ad esempio un bucket con tre oggetti che hanno i nomi delle chiavi seguenti. Sebbene questi siano archiviati senza un'organizzazione gerarchica fisica, è possibile dedurre la seguente struttura logica a cartelle a partire dai nomi delle chiavi:

- `sample1.jpg`: l'oggetto è nella root del bucket.
- `photos/2006/Jan/sample2.jpg`: l'oggetto è nella sottocartella `photos/2006/Jan`.
- `photos/2006/Feb/sample3.jpg`: l'oggetto è nella sottocartella `photos/2006/Feb`.

Nella console Amazon S3 è anche possibile creare una cartella in un bucket. Ad esempio, è possibile creare una cartella denominata `photos`. È possibile caricare gli oggetti nel bucket o nella cartella `photos` all'interno del bucket. Se si aggiunge l'oggetto `sample.jpg` al bucket, il nome della chiave

è `sample.jpg`. Se si carica l'oggetto nella cartella `photos`, il nome della chiave dell'oggetto è `photos/sample.jpg`.

Se si crea una struttura a cartelle nel bucket, occorre avere un documento di indice in ciascun livello. In ogni cartella, il documento di indice deve avere lo stesso nome, ad esempio, `index.html`. Quando un utente specifica un URL che si presenta come la ricerca di una cartella, la presenza o l'assenza di una barra finale determina il comportamento del sito Web. Ad esempio, il seguente URL, con barra finale, restituisce il documento di indice `photos/index.html`.

```
http://bucket-name.s3-website.Region.amazonaws.com/photos/
```

Tuttavia, se si esclude la barra finale dall'URL precedente, Amazon S3 cerca innanzitutto un oggetto `photos` nel bucket. Se non trova l'oggetto `photos`, cerca un documento indice, `photos/index.html`. Se questo documento viene trovato, Amazon S3 restituisce un messaggio `302 Found` e punta alla chiave `photos/`. Per le successive richieste `photos/`, Amazon S3 restituisce `photos/index.html`. Se il documento di indice non viene trovato, Amazon S3 restituisce un errore.

Configurazione di un documento indice

Per configurare un documento indice utilizzando la console S3, attieniti alla procedura seguente. Puoi anche configurare un documento indice utilizzando l'API REST, gli AWS SDK, o. AWS CLI AWS CloudFormation

> **Note**
>
> In un bucket abilitato al controllo delle versioni, puoi caricare più copie del file `index.html`, ma verrà risolta solo la versione più recente. Per ulteriori informazioni sulla funzione Controllo delle versioni S3, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Quando si abilita l'hosting statico di siti Web per il bucket, si immette il nome del documento di indice (ad esempi, **`index.html`**). Dopo aver abilitato l'hosting di siti Web statici per il bucket, si carica un file HTML con il nome del documento di indice nel bucket.

Per configurare il documento di indice

1. Creare un file `index.html`

Se non si dispone di un file `index.html`, è possibile utilizzare il seguente codice HTML per crearne 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. Salva il file indice in locale.

   Il nome del file del documento indice deve corrispondere esattamente al nome del documento indice immesso nella finestra di dialogo Hosting sito Web statico. Il nome del documento indice distingue tra maiuscole e minuscole. Ad esempio, se si immette `index.html` per il nome del documento Indice nella finestra di dialogo Hosting sito Web statico, anche il nome del file del documento indice deve essere `index.html` e non `Index.html`.

3. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

4. Nell'elenco S3 buckets (Bucket S3), scegliere il nome del bucket che si desidera utilizzare per ospitare un sito Web statico.

5. Abilitare l'hosting di siti Web statici per il bucket e inserire il nome esatto del documento di indice (ad esempi, `index.html`). Per ulteriori informazioni, consulta Abilitazione dell'hosting di siti Web.

   Dopo aver abilitato l'hosting di siti Web statici, procedere alla fase 6.

6. Per caricare il documento di indice nel bucket, eseguire una delle operazioni seguenti:

   - Trascinare e rilasciare il file di indice nell'elenco bucket della console.
   - Scegliere Upload (Carica) e seguire le istruzioni per scegliere e caricare il file di indice.

   Per step-by-step istruzioni, consultaCaricamento degli oggetti.

7. (Opzionale) Caricare altri contenuti del sito Web nel bucket.

Successivamente, è necessario impostare le autorizzazioni per l'accesso al sito Web. Per informazioni, consulta Impostazione delle autorizzazioni per l'accesso al sito Web.

È inoltre possibile configurare facoltativamente un documento di errore, laregistrazione del traffico Web o un reindirizzamento.

Configurazione di un documento di errore personalizzato

Dopo aver configurato il bucket come sito web statico, quando si verifica un errore, Amazon S3 restituisce un documento di errore HTML. È possibile configurare il bucket con un documento di errore personalizzato in modo che Amazon S3 restituisca tale documento quando si verifica un errore.

> **Note**
>
> In caso di errore, alcuni browser visualizzano il loro messaggio di errore, ignorando il documento di errore che restituisce Amazon S3. Ad esempio, quando si verifica un errore HTTP 404 Non trovato, Google Chrome potrebbe ignorare il documento di errore che Amazon S3 restituisce e visualizzare il suo errore.

Argomenti

Codici di risposta HTTP di Amazon S3

La seguente tabella elenca il sottoinsieme dei codici di risposta HTTP che Amazon S3 restituisce in caso di errore.

Codice di errore HTTP	Descrizione
301 Moved Permanently (301 Spostato definitiv amente)	Quando un utente invia una richiesta direttamente agli endpoint del sito web Amazon S3 (`http://s3-website.` *`Region`*`.amazonaws.com/`), Amazon S3 restituisce una risposta 301 Moved Permanently (301 Spostato

Codice di errore HTTP	Descrizione
	definitivamente) e reindirizza tali richieste a `https://aws.amazon.com/s3/` .
302 Found (302 Trovato)	Quando Amazon S3 riceve una richiesta per una chiave `x`, `http://bucket-name.s3-website.Region.amazonaws.com/x`, senza barra finale, cerca innanzitutto l'oggetto con nome della chiave `x`. Se l'oggetto non viene trovato, Amazon S3 stabilisce che la richiesta è per la sottocartella `x`, la reindirizza aggiungendo una barra finale e restituisce 302 Found (302 Trovato).
304 Not Modified (304 Non modificato)	Gli utenti Amazon S3 richiedono intestazioni `If-Modified-Since` , `If-Unmodified-Since` , `If-Match` e/o `If-None-Match` per stabilire se l'oggetto richiesto coincide con la copia memorizzata nella cache del client. Se l'oggetto coincide, l'endpoint del sito Web restituisce una risposta 304 Not Modified (304 Non modificato).
400 Malformed Request (400 Richiesta non corretta)	L'endpoint del sito Web restituisce una risposta 400 Malformed Request (400 Richiesta non corretta) quando un utente cerca di accedere a un bucket attraverso l'endpoint regionale sbagliato.
403 Forbidden (403 Non consentito)	L'endpoint del sito Web restituisce una risposta 403 Forbidden (403 Non consentito) quando la richiesta di un utente viene trasferita a un oggetto che non è pubblicamente leggibile. Il proprietario dell'oggetto deve rendere l'oggetto pubblicamente leggibile mediante una policy del bucket o un'ACL.

Codice di errore HTTP	Descrizione
404 Not Found (404 Non trovato)	L'endpoint del sito Web restituisce una risposta 404 Not Found (404 Non trovato) per i motivi seguenti: • Amazon S3 stabilisce che l'URL del sito web fa riferimento alla chiave di un oggetto che non esiste. • Amazon S3 deduce che la richiesta riguarda un documento di indice che non esiste. • Il bucket specificato nell'URL non esiste. • Il bucket specificato nell'URL esiste, ma non è configurato come sito Web. È possibile creare un documento personalizzato che viene restituito per 404 Not Found (404 Non trovato). Assicurarsi che il documento sia caricato nel bucket configurato come sito Web e che la configurazione di hosting del sito Web preveda l'utilizzo del documento. Per informazioni su come Amazon S3 interpreta l'URL come richiesta di un oggetto o di un documento di indice, consulta Configurazione di un documento indice.
500 Service Error (500 Errore servizio)	L'endpoint del sito Web restituisce una risposta 500 Service Error (500 Errore servizio) in caso di errore del server interno.
503 Service Unavailable (503 Servizio non disponibile)	L'endpoint del sito web restituisce una risposta 503 Service Unavailable (503 Servizio non disponibile) quando Amazon S3 stabilisce che occorre ridurre il tasso di richiesta.

Per ciascuno di questi errori, Amazon S3 restituisce un messaggio HTML predefinito. Di seguito è riportato un esempio di messaggio HTML che viene restituito per una risposta 403 Forbidden (403 Non consentito).

Configurazione di un documento di errore personalizzato

Quando configuri il bucket come sito Web statico, puoi fornire un documento di errore personalizzato contenente un messaggio di errore intuitivo e una guida aggiuntiva. Amazon S3 restituisce il documento di errore personalizzato solo per la classe dei codici di errore HTTP 4XX.

Per configurare un documento di errore personalizzato utilizzando la console S3, attenersi alla procedura riportata di seguito. Puoi anche configurare un documento di errore utilizzando l'API REST, AWS gli SDK AWS CLI, o AWS CloudFormation. Per ulteriori informazioni, consulta gli argomenti seguenti:

- PutBucketWebsitenel riferimento alle API di Amazon Simple Storage Service
- AWS::S3::Bucket WebsiteConfiguration nella Guida per l'utente di AWS CloudFormation
- put-bucket-website in Riferimento ai comandi AWS CLI

Quando abiliti l'hosting di siti Web statici per il tuo bucket, specifichi il nome del documento di errore (ad esempio, **`404.html`**). Dopo avere abilitato l'hosting di siti Web statici per il bucket, carichi un file HTML con il nome del documento di errore nel bucket.

Per configurare un documento di errore

1. Crea un documento di errore, ad esempio `404.html`.

2. Salva il file del documento di errore in locale.

   Il nome del documento di errore fa distinzione tra maiuscole e minuscole e deve corrispondere esattamente al nome immesso quando hai attivato l'hosting statico di siti Web. Ad esempio, se specifichi `404.html` per il nome del documento di errore nella finestra di dialogo Hosting sito Web statico, anche il nome file del documento di errore dovrà essere `404.html`.
3. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
4. Nell'elenco S3 buckets (Bucket S3), scegliere il nome del bucket che si desidera utilizzare per ospitare un sito Web statico.
5. Abilita l'hosting di siti Web statici per il bucket e inserisci il nome esatto del documento di errore (ad esempio, `404.html`). Per ulteriori informazioni, consulta Abilitazione dell'hosting di siti Web e Configurazione di un documento di errore personalizzato.

   Dopo aver abilitato l'hosting di siti Web statici, procedere alla fase 6.
6. Per caricare il documento di errore nel bucket, completa una delle operazioni riportate di seguito:

   - Trascina e rilascia il file del documento di errore nell'elenco dei bucket della console.
   - Scegliere Upload (Carica) e seguire le istruzioni per scegliere e caricare il file di indice.

   Per step-by-step istruzioni, consultaCaricamento degli oggetti.

Impostazione delle autorizzazioni per l'accesso al sito Web

Quando si configura un bucket come sito Web statico, se si desidera che il sito Web sia pubblico, è possibile concedere l'accesso per la lettura pubblica. Per consentire la lettura pubblica del bucket, occorre disabilitare le impostazioni di blocco dell'accesso pubblico al bucket e scrivere una policy del bucket che conceda l'accesso per la lettura pubblica. Se il bucket contiene oggetti che non appartengono al proprietario del bucket, potrebbe essere necessario aggiungere anche una lista di controllo degli accessi (ACL) dell'oggetto che concede a tutti l'accesso in lettura.

Se non vuoi disabilitare le impostazioni di blocco dell'accesso pubblico per il tuo bucket ma vuoi comunque che il tuo sito web sia pubblico, puoi creare una CloudFront distribuzione Amazon per servire il tuo sito web statico. Per ulteriori informazioni, consulta Velocizza il tuo sito Web con Amazon CloudFront o Usa una CloudFront distribuzione Amazon per servire un sito Web statico nella Amazon Route 53 Developer Guide.

> **Note**
>
> Nell'endpoint del sito web, se un utente richiede un oggetto che non esiste, Amazon S3 restituisce un codice di risposta HTTP `404 (Not Found)`. Se l'oggetto esiste ma la relativa autorizzazione di lettura non è stata concessa, l'endpoint del sito Web restituisce un codice di risposta HTTP `403 (Access Denied)`. L'utente può utilizzare il codice di risposta per capire se esiste un oggetto specifico. Per evitare questo tipo di comportamento, il supporto di siti Web per il bucket non deve essere abilitato.

Argomenti

Fase 1: modifica delle impostazioni dell'accesso pubblico ai blocchi Amazon S3

Per configurare un bucket esistente come sito web statico con accesso pubblico, devi modificare le impostazioni di blocco dell'accesso pubblico per il bucket. Potrebbe anche essere necessario modificare le impostazioni di blocca dell'accesso pubblico a livello di account. Amazon S3 applica la combinazione più restrittiva di impostazioni blocco di accesso pubblico a livello di account e a livello di bucket.

Ad esempio, se consenti l'accesso pubblico per un bucket ma lo blocchi a livello dell'account, Amazon S3 continuerà a bloccare l'accesso pubblico al bucket. In questo scenario, sarà necessario modificare le impostazioni di blocco dell'accesso pubblico a livello di bucket e di account. Per ulteriori informazioni, consulta Blocco dell'accesso pubblico allo storage Amazon S3.

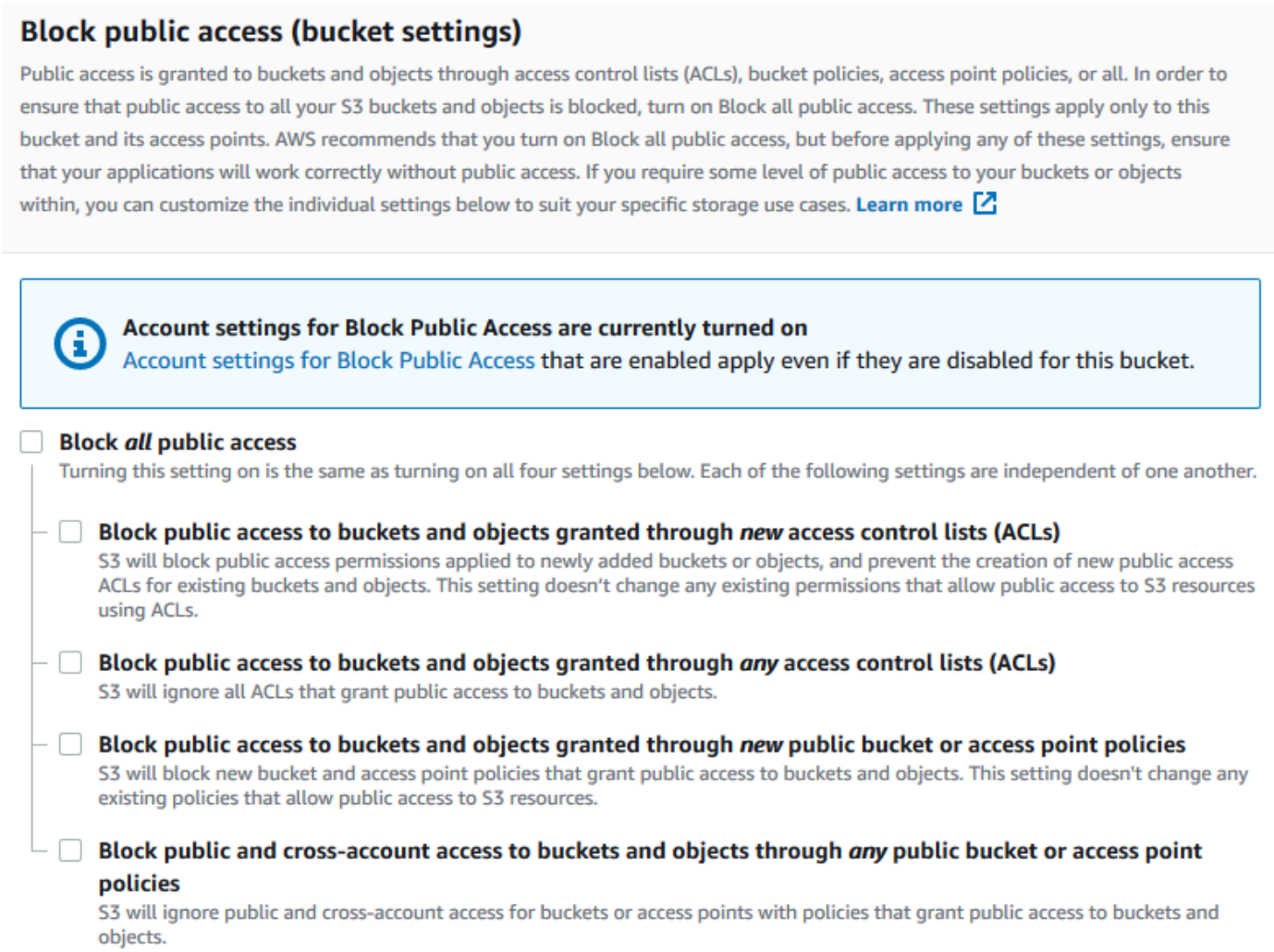
Per impostazione predefinita, Amazon S3 blocca l'accesso pubblico all'account e ai bucket. Per utilizzare un bucket per ospitare un sito Web statico, puoi seguire questa procedura per modificare le impostazioni di blocco dell'accesso pubblico:

> **Warning**
>
> Prima di completare questa fase, consulta Blocco dell'accesso pubblico allo storage Amazon S3 per confermare di avere compreso e accettato i rischi connessi alla concessione di un

accesso pubblico. Quando si disattivano le impostazioni di blocco dell'accesso pubblico per rendere pubblico il bucket, chiunque su Internet può accedere al bucket. Consigliamo di bloccare tutti gli accessi pubblici ai bucket.

1. Apri la console di Amazon S3 su https://console.aws.amazon.com/s3/.
2. Seleziona il nome del bucket configurato come sito Web statico.
3. Seleziona Autorizzazioni.
4. In Blocca accesso pubblico (impostazioni bucket), seleziona Modifica.
5. Deseleziona Blocca tutto l'accesso pubblico, quindi seleziona Salva modifiche.

> **Warning**
>
> Prima di completare questa fase, consulta Blocco dell'accesso pubblico allo storage Amazon S3 per confermare di avere compreso e accettato i rischi connessi alla concessione di un accesso pubblico. Quando si disattivano le impostazioni di blocco dell'accesso pubblico per rendere pubblico il bucket, chiunque su Internet può accedere al bucket. Consigliamo di bloccare tutti gli accessi pubblici ai bucket.



Amazon S3 disattiva le impostazioni di blocco dell'accesso pubblico per il tuo bucket. Per creare un sito web pubblico statico, potrebbe essere necessario modificare anche le impostazioni di blocco dell'accesso pubblico per l'account prima di aggiungere una policy del bucket. Se le impostazioni dell'account per il blocco dell'accesso pubblico sono attualmente attivate, verrà visualizzata una nota in Blocca accesso pubblico (impostazioni bucket).

Fase 2: aggiunta di una policy del bucket

Per rendere gli oggetti nel bucket pubblicamente leggibili, devi scrivere una policy del bucket che conceda a tutti l'autorizzazione `s3:GetObject`.

Dopo aver modificato le impostazioni di blocco dell'accesso pubblico S3, è possibile aggiungere una policy del bucket per concedere l'accesso pubblico in lettura al bucket. Quando concedi l'accesso pubblico in lettura, chiunque su Internet può accedere al bucket.

> **Important**
>
> La policy seguente è solo un esempio e consente l'accesso completo ai contenuti del bucket. Prima di continuare con questa fase, esamina l'argomento relativo a come proteggere i file nel bucket Amazon S3 per assicurarti di comprendere le best practice per la protezione dei file nel bucket S3 e i rischi connessi alla concessione dell'accesso pubblico .

1. In Bucket, scegli il nome del bucket.
2. Seleziona Autorizzazioni.
3. In Policy del bucket, seleziona Modifica.
4. Per concedere l'accesso in lettura pubblico al sito Web, copiare la policy del bucket seguente e incollarla in Editor della policy del bucket.

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

5. Aggiorna `Resource` al tuo nome bucket.

   Nella policy del bucket dell'esempio precedente, *`Bucket-Name`* è un segnaposto per il nome del bucket. Per utilizzare questa policy di bucket con il proprio bucket, è necessario aggiornare il nome in modo che corrisponda al bucket.

6. Seleziona Salva modifiche.

   Viene visualizzato un messaggio che indica che la policy del bucket è stata aggiunta correttamente.

   Se viene visualizzato l'errore `Policy has invalid resource`, conferma che il nome del bucket nella policy di bucket corrisponde al nome del bucket. Per informazioni sull'aggiunta di una policy del bucket, consulta In che modo aggiungere una policy del bucket S3?

   Se viene visualizzato un messaggio di errore e non è possibile salvare la policy di bucket, controlla le impostazioni di blocco dell'accesso pubblico all'account e al bucket per confermare che consenti l'accesso pubblico al bucket.

Liste di controllo accessi dell'oggetto

Puoi utilizzare una policy del bucket per concedere l'autorizzazione in lettura pubblica per gli oggetti. Tuttavia, la policy del bucket si applica solo agli oggetti appartenenti al proprietario del bucket. Se il bucket contiene oggetti che non appartengono al proprietario del bucket, quest'ultimo deve utilizzare la lista di controllo degli accessi (ACL) per concedere l'autorizzazione READ pubblica per tali oggetti.

S3 Proprietà dell'oggetto è un'impostazione a livello di bucket Amazon S3 che è possibile utilizzare per controllare la proprietà degli oggetti caricati nel bucket e per disabilitare o abilitare le liste di controllo degli accessi (ACL). Per impostazione predefinita, Proprietà dell'oggetto è impostata su Proprietario del bucket applicato e tutte le liste di controllo degli accessi (ACL) sono disabilitate. Quando le ACL sono disabilitate, il proprietario del bucket dispone di tutti gli oggetti nel bucket e gestisce l'accesso ad essi in maniera esclusiva utilizzando policy di gestione dell'accesso.

La maggior parte degli attuali casi d'uso in Amazon S3 non richiede più l'uso delle ACL. È consigliabile mantenere le ACL disabilitate, tranne nelle circostanze in cui è necessario controllare individualmente l'accesso per ciascun oggetto. Con le ACL disabilitate, puoi utilizzare le policy per controllare l'accesso a tutti gli oggetti nel bucket, a prescindere da chi ha caricato gli oggetti nel bucket. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione degli ACL per il bucket.

> **Important**
>
> Se il bucket utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto S3, è necessario utilizzare le policy per concedere l'accesso al bucket e agli oggetti in esso contenuti. Con l'impostazione Proprietario del bucket applicato abilitata, le richieste per

impostare liste di controllo degli accessi (ACL) e aggiornare le ACL non vanno a buon fine e restituiscono il codice di errore `AccessControlListNotSupported`. Le richieste di lettura delle ACL sono ancora supportate.

Per rendere un oggetto pubblicamente leggibile mediante un ACL, occorre concedere l'autorizzazione READ al gruppo `AllUsers`, come illustrato nel seguente elemento "grant". Aggiungere questo elemento "grant" all'ACL dell'oggetto. Per informazioni sulla gestione delle ACL, consulta Panoramica delle liste di controllo accessi (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>
```

(Facoltativo) Registrazione del traffico Web

Facoltativamente puoi abilitare la registrazione dell'accesso al server Amazon S3 per un bucket configurato come sito web statico. La registrazione degli accessi al server fornisce record dettagliati per le richieste che sono effettuate al bucket. Per ulteriori informazioni, consulta Registrazione delle richieste con registrazione dell'accesso al server. Se prevedi di utilizzare Amazon CloudFront per velocizzare il tuo sito Web, puoi anche utilizzare CloudFront la registrazione. Per ulteriori informazioni, consulta Configurazione e utilizzo dei log di accesso nella Amazon CloudFront Developer Guide.

Per abilitare la registrazione dell'accesso al server per il bucket del sito Web statico

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nella stessa regione in cui è stato creato il bucket configurato come sito Web statico, creare un bucket per la registrazione, ad esempio `logs.example.com`.
3. Creare una cartella per i file di registrazione degli accessi al server (ad esempio, `logs`).
4. (Facoltativo) Se desideri utilizzarlo CloudFront per migliorare le prestazioni del tuo sito Web, crea una cartella per i file di CloudFront registro (ad esempio,`cdn`).

   Per ulteriori informazioni, consulta Velocizza il tuo sito Web con Amazon CloudFront.

5. Nell'elenco Bucket, seleziona il nome del bucket.
6. Scegliere Properties (Proprietà).
7. In Registrazione accesso server, seleziona Modifica.
8. Scegli Enable (Abilita).
9. In Bucket di destinazione, seleziona la destinazione del bucket e della cartella per i log di accesso al server:

   - Individua la cartella e il percorso del bucket:

     1. Seleziona Sfoglia S3.
     2. Scegli il nome del bucket, quindi seleziona la cartella dei log.
     3. Seleziona Scegli percorso.

   - Specifica il percorso del bucket S3, ad esempio, **`s3://logs.example.com/logs/`**.

10. Seleziona Salva modifiche.

    Nel bucket di log, ora puoi accedere ai tuoi log. Amazon S3 scrive i log di accesso del sito web nel bucket log ogni due ore.

(Facoltativo) Configurazione del reindirizzamento di una pagina Web

Se il bucket Amazon S3 è configurato per l'hosting di siti Web statici, è possibile configurare i reindirizzamenti per il bucket o gli oggetti in esso contenuti. Per configurare il reindirizzamento sono disponibili le opzioni riportate di seguito.

Argomenti

- Reindirizzamento delle richieste per l'endpoint del sito Web del bucket a un altro bucket o dominio
- Configurazione delle regole di reindirizzamento per utilizzare i reindirizzamenti condizionali avanzati
- Reindirizzamento delle richieste per un oggetto

Reindirizzamento delle richieste per l'endpoint del sito Web del bucket a un altro bucket o dominio

È possibile reindirizzare tutte le richieste a un endpoint di sito Web per un bucket a un altro bucket o a un dominio. Se vengono reindirizzate tutte le richieste, qualsiasi richiesta effettuata all'endpoint del sito Web viene reindirizzata al bucket o al dominio specificato.

Ad esempio, se il dominio root è `example.com` e desideri servire richieste sia per `http://example.com` che per `http://www.example.com`, puoi creare due bucket denominati `example.com` e `www.example.com`. Successivamente, mantenere il contenuto nel bucket `example.com` e configurare l'altro bucket `www.example.com` per reindirizzare tutte le richieste al bucket `example.com`. Per ulteriori informazioni, consulta Configurazione di un sito Web statico utilizzando un nome di dominio personalizzato.

Per reindirizzare le richieste per un endpoint di un sito Web bucket

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. In Bucket seleziona il nome del bucket da cui desideri reindirizzare le richieste (ad esempio, `www.example.com`).
3. Scegliere Properties (Proprietà).
4. In Hosting di siti Web statici, seleziona Modifica.
5. Seleziona Reindirizza richieste per un oggetto.
6. Nella casella Nome host specifica l'endpoint del sito Web per il bucket o il dominio personalizzato.

   Ad esempio, se il reindirizzamento è effettuato verso un indirizzo di dominio root, digita **`example.com`**.
7. Per Protocollo, seleziona il protocollo per le richieste reindirizzate (nessuno,http o https).

   Se non si specifica un protocollo, l'opzione predefinita è nessuno.
8. Seleziona Salva modifiche.

Configurazione delle regole di reindirizzamento per utilizzare i reindirizzamenti condizionali avanzati

Con le regole di reindirizzamento avanzato, è possibile instradare le richieste in modo condizionale in base ai nomi delle chiavi degli oggetti specifici, ai prefissi nella richiesta o ai codici di risposta. Si supponga ad esempio di eliminare o rinominare un oggetto nel bucket. È possibile aggiungere una regola di routing che reindirizza la richiesta a un altro oggetto. Se si desidera rendere una cartella non disponibile, è possibile aggiungere una regola di routing per reindirizzare la richiesta a un'altra pagina Web. Inoltre, è possibile aggiungere una regola di routing per gestire le condizioni di errore instradando le richieste che restituiscono l'errore a un altro dominio dove viene elaborato l'errore.

Quando abiliti l'hosting di siti Web statici per il tuo bucket, puoi specificare facoltativamente regole di reindirizzamento avanzate. Amazon S3 ha un limite di 50 regole di routing per configurazione di sito web. Se sono necessarie più di 50 regole di routing, è possibile utilizzare l'instradamento degli oggetti. Per ulteriori informazioni, consulta Utilizzo della console S3.

Per ulteriori informazioni sulla configurazione delle regole di routing utilizzando l'API REST, consulta PutBucketWebsiteAmazon Simple Storage Service API Reference.

> **Important**
>
> Per creare regole di reindirizzamento nella nuova console Amazon S3, è necessario utilizzare JSON. Per gli esempi JSON, consulta Esempi regole di reindirizzamento.

Per configurare le regole di reindirizzamento per un sito Web statico

Per aggiungere le regole di reindirizzamento per un bucket che ha già abilitato l'hosting di siti Web statici, attieniti alla seguente procedura.

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket, seleziona il nome di un bucket configurato come sito Web statico.
3. Scegliere Properties (Proprietà).
4. In Hosting di siti Web statici, seleziona Modifica.
5. Nella casella Redirection rules (Regole di reindirizzamento), immettere le regole di reindirizzamento in JSON.

Nella console S3 descrivi le regole utilizzando JSON. Per gli esempi JSON, consulta Esempi regole di reindirizzamento. Amazon S3 ha un limite di 50 regole di routing per configurazione di sito web.

6. Seleziona Salva modifiche.

Elementi regola instradamento

Di seguito è riportata la sintassi generale per definire le regole di routing in una configurazione di un sito Web in XML. Per configurare le regole di reindirizzamento nella nuova console S3, è necessario utilizzare JSON. Per gli esempi JSON, consulta Esempi regole di reindirizzamento.

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.
```

Nella seguente tabella sono descritti gli elementi della regola di routing.

Nome	Descrizione
`RoutingRules`	Container per una raccolta di elementi `RoutingRule` .
`RoutingRule`	Una regola che stabilisce una condizione e il reindirizzamento che viene applicato quando la condizione è soddisfatta. Condizione: • Un container `RoutingRules` deve contenere almeno una regola di routing.

Nome	Descrizione
`Condition`	Container per descrivere una condizione che deve essere soddisfatta per l'applicazione del reindirizzamento specificato. Se la regola di routing non include una condizione, la regola viene applicata a tutte le richieste.
`KeyPrefixEquals`	Il prefisso del nome della chiave dell'oggetto da cui vengono reindirizzate le richieste. `KeyPrefixEquals` è obbligatorio se `HttpErrorCodeReturnedEquals` non è specificato. Se `KeyPrefixEquals` e `HttpErrorCodeReturnedEquals` sono specificati, devono essere entrambi veri perché la condizione sia soddisfatta.
`HttpErrorCodeReturnedEquals`	Il codice di errore HTTP che deve corrispondere perché il reindirizzamento venga applicato. Se si verifica un errore e se il codice di errore corrisponde a questo valore, il reindirizzamento specificato viene applicato. `HttpErrorCodeReturnedEquals` è obbligatorio se `KeyPrefixEquals` non è specificato. Se `KeyPrefixEquals` e `HttpErrorCodeReturnedEquals` sono specificati, devono essere entrambi veri perché la condizione sia soddisfatta.
`Redirect`	Elemento del container che fornisce istruzioni per il reindirizzamento della richiesta. È possibile reindirizzare le richieste a un altro host o a un'altra pagina oppure specificare un altro protocollo da utilizzare. Un `RoutingRule` deve avere un elemento `Redirect`. Un elemento `Redirect` deve contenere almeno uno dei seguenti elementi di pari livello: `Protocol`, `HostName`, `ReplaceKeyPrefixWith`, `ReplaceKeyWith` o `HttpRedirectCode`.

Nome	Descrizione
`Protocol`	Il protocollo, `http` o `https`, da utilizzare nell'intestazione `Location` che viene restituita nella risposta. Se viene fornito uno degli elementi di pari livello, `Protocol` non è necessario.
`HostName`	Il nome dell'host da utilizzare nell'intestazione `Location` che viene restituita nella risposta. Se viene fornito uno degli elementi di pari livello, `HostName` non è necessario.
`ReplaceKeyPrefixWith`	Il prefisso del nome della chiave dell'oggetto che sostituisce il valore di `KeyPrefixEquals` nella richiesta di reindirizzamento. Se viene fornito uno degli elementi di pari livello, `ReplaceKeyPrefixWith` non è necessario. Può essere fornito solo se `ReplaceKeyWith` non è fornito.
`ReplaceKeyWith`	La chiave dell'oggetto da utilizzare nell'intestazione `Location` che viene restituita nella risposta. Se viene fornito uno degli elementi di pari livello, `ReplaceKeyWith` non è necessario. Può essere fornito solo se `ReplaceKeyPrefixWith` non è fornito.
`HttpRedirectCode`	Il codice di reindirizzamento HTTP da utilizzare nell'intestazione `Location` che viene restituita nella risposta. Se viene fornito uno degli elementi di pari livello, `HttpRedirectCode` non è necessario.

Esempi regole di reindirizzamento

Gli esempi seguenti illustrano le comuni attività di reindirizzamento:

> **Important**
>
> Per creare regole di reindirizzamento nella nuova console Amazon S3, è necessario utilizzare JSON.

Example 1: reindirizzamento dopo la ridenominazione del prefisso di una chiave

Si supponga che il bucket contenga i seguenti oggetti:

- index.html
- docs/article1.html
- docs/article2.html

Si decide di rinominare la cartella da `docs/` a `documents/`. Dopo aver apportato questa modifica, occorre reindirizzare le richieste del prefisso `docs/` verso `documents/`. Ad esempio, la richiesta di `docs/article1.html` sarà reindirizzata a `documents/article1.html`.

In questo caso, si aggiunge la seguente regola di routing alla configurazione del sito 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: reindirizzamento delle richieste di una cartella eliminata verso una pagina

Si supponga l'eliminazione della cartella `images/` (ovvero, l'eliminazione di tutti gli oggetti con prefisso della chiave `images/`). È possibile aggiungere una regola di routing che reindirizza le richieste di qualsiasi oggetto con prefisso della chiave `images/` verso una pagina denominata `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: reindirizzamento per un errore HTTP

Si supponga che quando non viene trovato un oggetto richiesto, si desidera reindirizzare le richieste a un'istanza Amazon Elastic Compute Cloud (Amazon EC2). Aggiungere una regola di reindirizzamento in modo tale che, quando viene restituito un codice di stato HTTP 404 (Non trovato), il visitatore del sito venga reindirizzato a un'istanza Amazon EC2 che gestisca la richiesta.

Il seguente esempio riporta nel reindirizzamento anche il prefisso della chiave dell'oggetto `report-404/`. Ad esempio, se la richiesta di una pagina `ExamplePage.html` restituisce un errore HTTP 404, la richiesta viene reindirizzata a una pagina `report-404/ExamplePage.html` sull'istanza Amazon EC2 specificata. Se non sono presenti regole di routing e si verifica l'errore HTTP 404, viene restituito il documento di errore specificato nella configurazione.

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>
```

Reindirizzamento delle richieste per un oggetto

Puoi reindirizzare le richieste di un oggetto a un altro oggetto o URL impostando la posizione di reindirizzamento del sito Web nei metadati dell'oggetto. Si imposta il reindirizzamento aggiungendo la proprietà `x-amz-website-redirect-location` ai metadata dell'oggetto. Nella console Nella console Amazon S3, la Posizione di reindirizzamento del sito Web si imposta nei metadati dell'oggetto. Se utilizzi l'API Amazon S3, hai impostato `x-amz-website-redirect-location`. Il sito Web interpreta quindi l'oggetto come reindirizzamento 301.

Per reindirizzare una richiesta a un altro oggetto, si imposta la posizione di reindirizzamento sulla chiave dell'oggetto di destinazione. Per reindirizzare una richiesta a un URL esterno, si imposta la posizione di reindirizzamento sull'URL desiderato. Per ulteriori informazioni sui metadati degli oggetti, consulta Metadata di oggetti definiti dal sistema.

Quando si imposta il reindirizzamento di una pagina, è possibile mantenere o eliminare il contenuto dell'oggetto di origine. Ad esempio, se nel bucket è presente un oggetto `page1.html`, è possibile reindirizzare qualsiasi richiesta per questa pagina a un altro oggetto `page2.html`. Sono disponibili due opzioni:

- Mantenere il contenuto dell'oggetto `page1.html` e reindirizzare le richieste per la pagina.
- Eliminare il contenuto di `page1.html` e caricare un oggetto a zero byte denominato `page1.html` per sostituire l'oggetto esistente e reindirizzare le richieste per la pagina.

Utilizzo della console S3

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket, seleziona il nome di un bucket configurato come sito Web statico (ad esempio, `example.com`).
3. In Oggetti, seleziona l'oggetto.
4. Seleziona Operazioni, quindi Modifica metadati.
5. Seleziona Metadati.
6. Seleziona Aggiungi metadati.
7. In Tipo, seleziona Definito dal sistema.

8. In Key, scegli x-amz-website-redirect -location.
9. In Valore, immettere il nome della chiave dell'oggetto a cui si desidera reindirizzare, ad esempio `/page2.html`.

   Per un altro oggetto nello stesso bucket, il prefisso `/` nel valore è obbligatorio. È possibile inoltre impostare il valore su un URL esterno, ad esempi, `http://www.example.com`.
10. Seleziona Modifica metadati.

Utilizzo di REST API

Le seguenti operazioni API Amazon S3 supportano l'intestazione `x-amz-website-redirect-location` nella richiesta. Amazon S3 archivia il valore dell'intestazione nei metadati dell'oggetto come `x-amz-website-redirect-location`.

- PUT Object
- Avvio del caricamento in più parti
- POST Object
- PUT Object - Copy

Un bucket configurato per l'hosting di siti Web presenta sia l'endpoint del sito Web che l'endpoint REST. La richiesta di una pagina configurata come reindirizzamento 301 può generare i seguenti risultati, a seconda dell'endpoint della richiesta:

- Endpoint del sito web specifico per regione: Amazon S3 reindirizza la richiesta della pagina in base al valore della proprietà `x-amz-website-redirect-location`.
- Endpoint REST: Amazon S3 non reindirizza la richiesta della pagina. Restituisce l'oggetto richiesto.

Per ulteriori informazioni sugli endpoint, consulta Differenze chiave tra un endpoint del sito Web e un endpoint REST API.

Quando si imposta il reindirizzamento di una pagina, è possibile mantenere o eliminare il contenuto dell'oggetto. Supponi, ad esempio, di avere un oggetto `page1.html` nel bucket.

- Per mantenere il contenuto di `page1.html` e reindirizzare solo le richieste della pagina, è possibile inviare una richiesta PUT Object - Copy per creare un nuovo oggetto `page1.html` che utilizzi l'oggetto `page1.html` esistente come origine. Nella richiesta, si imposta l'intestazione `x-amz-`

`website-redirect-location`. Al completamento della richiesta, si ottiene la pagina originale con contenuto invariato, ma Amazon S3 reindirizza qualsiasi richiesta della pagina alla posizione di reindirizzamento specificata.

- Per eliminare il contenuto dell'oggetto `page1.html` e reindirizzare le richieste della pagina, è possibile inviare una richiesta PUT Object per caricare un oggetto da zero byte con la stessa chiave dell'oggetto: `page1.html`. Nella richiesta PUT, si imposta `x-amz-website-redirect-location` per `page1.html` sul nuovo oggetto. Al completamento della richiesta, `page1.html` non ha contenuto e le richieste vengono reindirizzate alla posizione specificata da `x-amz-website-redirect-location`.

Quando si recupera l'oggetto tramite l'operazione GET Object, insieme ad altri metadati dell'oggetto, Amazon S3 restituisce nella risposta l'intestazione `x-amz-website-redirect-location`.

Sviluppo con Amazon S3

In questa sezione sono riportati gli argomenti relativi agli sviluppatori per l'uso di Amazon S3. Per ulteriori informazioni, consulta gli argomenti riportati di seguito.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Argomenti

Esecuzione di richieste

Amazon S3 è un servizio REST. È possibile inviare le richieste ad Amazon S3 utilizzando le librerie wrapper dell'API REST o dell'SDK AWS (consulta Codice di esempio e librerie) che eseguono il wrapping dell'API REST di Amazon S3 sottostante, semplificando le attività di programmazione.

Ogni interazione con Amazon S3 è autenticata o anonima. L'autenticazione è un processo teso a verificare l'identità del richiedente che cerca di accedere a un prodotto Amazon Web Services (AWS). Le richieste autenticate devono includere un valore di firma che autentichi il mittente della richiesta. Il valore di firma è, in parte, generato dalle chiavi di accesso AWS del richiedente (ID chiave di accesso e chiave di accesso segreta). Per ulteriori informazioni su come ottenere le chiavi di accesso, consulta Come ottenere le credenziali di sicurezza? in Riferimenti generali di AWS.

Se si utilizza l'SDK AWS, le librerie calcolano la firma dalla chiave fornita. Tuttavia, se vengono effettuate chiamate API REST direttamente dall'applicazione in uso, è necessario scrivere il codice per calcolare la firma e aggiungerlo alla richiesta.

Argomenti

Le chiavi di accesso

Le sezioni seguenti esaminano i tipi di chiavi di accesso che è possibile utilizzare per effettuare richieste autenticate.

Chiavi di accesso di Account AWS

Le chiavi di accesso a un account consentono l'accesso completo alle risorse AWS di proprietà dell'account. Di seguito vengono illustrati alcuni esempi di chiavi di accesso:

- ID chiave di accesso (una stringa alfanumerica di 20 caratteri). Ad esempio: AKIAIOSFODNN7EXAMPLE
- Secret Access Key (una stringa di 40 caratteri). Ad esempio: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

L'ID chiave di accesso identifica in maniera univoca un Account AWS. È possibile utilizzare queste chiavi di accesso per inviare le richieste autenticate a Amazon S3.

Chiavi di accesso utente IAM

È possibile creare un Account AWS aziendale; tuttavia, ci potrebbero essere alcuni dipendenti che hanno necessità di accedere alle risorse AWS dell'organizzazione. La condivisione delle chiavi di accesso Account AWS riduce la sicurezza mentre la creazione di singoli Account AWS per ogni dipendente potrebbe essere poco pratica. Inoltre, non è facile condividere risorse quali i bucket e gli oggetti in quanto sono di proprietà di diversi account. Per condividere le risorse, è necessario assegnare le autorizzazioni, il che comporta un lavoro aggiuntivo.

In questi scenari, è possibile utilizzare AWS Identity and Access Management (IAM) per creare utenti all'interno dell'Account AWS che dispongano di chiavi di accesso proprie e collegare le policy utente

IAM che assegnino le opportune autorizzazioni di accesso alle risorse a tali utenti. Per gestire meglio tali utenti, IAM consente la creazione di gruppi di utenti e l'assegnazione di autorizzazioni a livello di gruppo valide per tutti gli utenti presenti in quel gruppo.

Questi sono denominati utenti IAM, creati e gestiti all'interno di AWS. L'account padre verifica la capacità di un utente di accedere ad AWS. Qualsiasi risorsa creata da un utente IAM ricade sotto il controllo dell' Account AWS padre e viene pagata da quest'ultimo. Questi utenti IAM possono inviare richieste autenticate a Amazon S3 utilizzando le loro credenziali di sicurezza. Per ulteriori informazioni su creazione e gestione degli utenti all'interno dell'Account AWS, visita la pagina dei dettagli del prodotto AWS Identity and Access Management.

Credenziali di sicurezza temporanee

Oltre a creare utenti IAM che dispongono di chiavi di accesso proprie, IAM consente anche di assegnare credenziali di sicurezza temporanee (chiavi di accesso temporanee e un token di sicurezza) a qualsiasi utente IAM, per consentire l'accesso ai servizi e alle risorse AWS. È inoltre possibile gestire gli utenti nel sistema in uso al di fuori di AWS. Tali risorse sono denominate utenti federati. Inoltre, gli utenti possono essere applicazioni create per accedere alle risorse AWS.

IAM fornisce l'API AWS Security Token Service per la richiesta di credenziali di sicurezza temporanee. Per richiedere le credenziali, è possibile utilizzare l'API STS AWS oppure l'SDK AWS. L'API restituisce le credenziali di sicurezza (ID chiave di accesso e Secret Access Key) e un token di sicurezza. Queste credenziali sono valide solo per la durata specificata al momento della richiesta. L'ID chiave di accesso e la chiave segreta si utilizzano nello stesso modo in cui vengono impiegati quando si inviano le richieste usando l'Account AWS o le chiavi di accesso dell'utente IAM. Inoltre, è necessario includere il token in ogni richiesta inviata a Amazon S3.

Un utente IAM può richiedere queste credenziali di sicurezza temporanee per uso personale oppure per passarle agli utenti federati o alle applicazioni. Quando si effettua la richiesta di credenziali di sicurezza per gli utenti federati, è necessario fornire un nome utente e una policy IAM che definisce le autorizzazioni che si vuole associare a tali credenziali di sicurezza. L'utente federato non può avere un numero di autorizzazioni superiore a quello di cui dispone l'utente IAM padre che ha richiesto le credenziali di sicurezza.

È possibile utilizzare queste credenziali di sicurezza temporanee quando si effettuano richieste ad Amazon S3. Le librerie API calcolano il valore di firma necessario utilizzando tali credenziali per autenticare la richiesta dell'utente. In caso di invio di richieste tramite l'utilizzo di credenziali scadute, Amazon S3 rifiuta la richiesta.

Per informazioni sulla firma delle richieste mediante l'utilizzo di credenziali di sicurezza temporanee nelle richieste API REST, consulta Firma e autenticazione delle richieste REST. Per informazioni sull'invio di richieste utilizzando gli SDK AWS, consulta Esecuzione di richieste tramite gli SDK AWS.

Per ulteriori informazioni sul supporto IAM per le credenziali di sicurezza temporanee, consulta la sezione relativa alle credenziali di sicurezza temporanee nella Guida per l'utente di IAM.

Per maggiore sicurezza, è possibile richiedere l'autenticazione a più fattori (MFA) quando si effettua l'accesso alle risorse Amazon S3 configurando una policy del bucket. Per informazioni, consulta Richiesta dell'autenticazione a più fattori (MFA). Una volta richiesta l'MFA per accedere alle risorse Amazon S3, il solo modo per accedervi è fornendo le credenziali temporanee che vengono create con una chiave MFA. Per ulteriori informazioni, consulta la pagina dei dettagli Autenticazione a più fattori (MFA) di AWS e Configurazione dell'accesso alle API protetto da MFA nella Guida per l'utente di IAM.

Endpoint della richiesta

Le richieste REST vengono inviate all'endpoint predefinito del servizio. Per un elenco di tutti i servizi AWS e degli endpoint corrispondenti, consultare la sezione relativa a regioni ed endpoint nella Riferimenti generali di AWS.

Esecuzione di richieste ad Amazon S3 su IPv6

Amazon Simple Storage Service (Amazon S3) consente di accedere ai bucket S3 tramite l'Internet Protocol versione 6 (IPv6), oltre al protocollo IPv4. Gli endpoint dual-stack Amazon S3; supportano le richieste ai bucket S3 su IPv6 e IPv4. Non sono previsti costi aggiuntivi per l'accesso ad Amazon S3 su IPv6. Per ulteriori informazioni sui prezzi, consulta Prezzi di Amazon S3.

Argomenti

- Nozioni di base sull'esecuzione di richieste su IPv6
- Utilizzo degli indirizzi IPv6 nelle policy IAM
- Test di compatibilità degli indirizzi IP
- Utilizzo degli endpoint dual-stack Amazon S3

Nozioni di base sull'esecuzione di richieste su IPv6

Per effettuare una richiesta a un bucket S3 su IPv6, è necessario utilizzare un endpoint dual-stack. Nella sezione seguente viene descritto come effettuare richieste su IPv6 utilizzando gli endpoint dual-stack.

Di seguito sono descritti alcuni punti da prendere in considerazione prima di tentare di accedere a un bucket su IPv6:

- Il client e la rete che eseguono l'accesso al bucket devono essere abilitati a utilizzare IPv6.
- Per l'accesso a IPv6 sono supportate sia le richieste in stile percorso sia quelle in stile hosting virtuale. Per ulteriori informazioni, consulta Endpoint dual-stack Amazon S3.
- Se utilizzi il filtraggio degli indirizzi IP di origine nelle tue policy utente o bucket AWS Identity and Access Management (IAM), devi aggiornare le policy per includere gli intervalli di indirizzi IPv6. Per ulteriori informazioni, consulta Utilizzo degli indirizzi IPv6 nelle policy IAM.
- Quando si utilizza IPv6, i file di log degli accessi al server generano indirizzi IP in un formato IPv6. È necessario aggiornare il software, gli script e gli strumenti esistenti utilizzati per analizzare i file di log di Amazon S3; affinché possano analizzare gli indirizzi `Remote IP` in formato IPv6. Per ulteriori informazioni, consulta Formato del log di accesso al server Amazon S3 e Registrazione delle richieste con registrazione dell'accesso al server.

> **Note**
>
> Se si verificano problemi relativi alla presenza di indirizzi IPv6 nei file di log, contatta AWS Support.

Esecuzione di richieste su IPv6 tramite gli endpoint dual-stack

È possibile effettuare richieste con chiamate all'API di Amazon S3 tramite IPv6 utilizzando gli endpoint dual-stack. Le operazioni dell'API di Amazon S3 vengono eseguite nello stesso modo sia che tu acceda a Amazon S3 su IPv6 che su IPv4. Anche le prestazioni dovrebbero essere le stesse.

Quando si utilizza l'API REST, è possibile accedere direttamente a un endpoint dual-stack. Per ulteriori informazioni, consulta Endpoint dual-stack.

Quando usi AWS Command Line Interface (AWS CLI) e gli AWS SDK, puoi usare un parametro o un flag per passare a un endpoint dual-stack. È inoltre possibile specificare l'endpoint dual-stack direttamente come sostituzione dell'endpoint Amazon S3 nel file di configurazione.

È possibile utilizzare un endpoint dual-stack per accedere a un bucket su IPv6 da uno qualsiasi dei seguenti elementi:

- AWS CLI IIUtilizzo degli endpoint dual-stack di AWS CLI, vedi.
- Gli AWS SDK, vediUtilizzo degli endpoint dual-stack dagli SDK AWS.
- API REST, consulta Esecuzione di richieste a endpoint Dual-Stack utilizzando l'API REST.

Caratteristiche non disponibili su IPv6

La seguente funzionalità non è attualmente supportata quando si accede a un bucket S3 su IPv6: sito Web statico hosting da un bucket S3.

Utilizzo degli indirizzi IPv6 nelle policy IAM

Prima di tentare di accedere a un bucket tramite IPv6-S3, è necessario assicurarsi che le policy utente IAM o del bucket IPv6 utilizzate per il filtro degli indirizzi IP siano aggiornate per includere gli intervalli di indirizzi IPv6. Se non aggiorni le policy di filtro degli indirizzi IP per la gestione degli indirizzi IPv6, i client possono perdere oppure ottenere in modo errato l'accesso al bucket quando iniziano a utilizzare IPv6. Per ulteriori informazioni sulla gestione delle autorizzazioni di accesso con IAM, consulta Identity and Access Management per Amazon S3.

Le policy IAM che filtrano gli indirizzi IP utilizzano gli operatori di condizione degli indirizzi IP. La seguente policy del bucket identifica l'intervallo 54.240.143.* degli indirizzi IPv4 consentiti tramite gli operatori di condizione degli indirizzi IP. A qualsiasi indirizzo IP non incluso in questo intervallo verrà negato l'accesso al bucket (`examplebucket`). Poiché tutti gli indirizzi IPv6 non sono inclusi nell'intervallo consentito, questa policy impedisce agli indirizzi IPv6 di accedere 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"}
      }
    }
  ]
}
```

È possibile modificare l'elemento `Condition` della policy del bucket per consentire entrambi gli intervalli di indirizzi IPv4 (`54.240.143.0/24`) e IPv6 (`2001:DB8:1234:5678::/64`), come mostrato nell'esempio che segue. È possibile utilizzare lo stesso tipo di blocco `Condition` mostrato nell'esempio per aggiornare sia le policy del bucket sia le policy utente IAM.

```
       "Condition": {
         "IpAddress": {
            "aws:SourceIp": [
              "54.240.143.0/24",
               "2001:DB8:1234:5678::/64"
             ]
          }
        }
```

Prima di utilizzare IPv6, è necessario aggiornare tutte le policy del bucket e utente IAM pertinenti che utilizzano il filtro degli indirizzi IP per consentire gli intervalli di indirizzi IPv6. Ti consigliamo di aggiornare le policy IAM con gli intervalli di indirizzi IPv6 dell'organizzazione oltre agli intervalli di indirizzi IPv4 esistenti. Per un esempio di una policy del bucket che consenta l'accesso sia su IPv6 sia su IPv4, consulta Limitare l'accesso a indirizzi IP specifici.

È possibile esaminare le policy utente IAM tramite la console IAM all'indirizzo https://console.aws.amazon.com/iam/. Per ulteriori informazioni su IAM, consulta la Guida per l'utente di IAM. Per informazioni sulla modifica delle policy dei bucket S3, consulta Aggiunta di una policy di bucket utilizzando la console di Amazon S3.

Test di compatibilità degli indirizzi IP

Se utilizzi Linux/Unix o Mac OS X, puoi verificare se è possibile accedere a un endpoint dual-stack su IPv6 tramite il comando `curl`, come mostrato nell'esempio seguente:

Example

```
curl -v  http://s3.dualstack.us-west-2.amazonaws.com/
```

Vengono restituite informazioni simili a quelle indicate nell'esempio seguente. Se la connessione è stata eseguita su IPv6, l'indirizzo IP connesso sarà un indirizzo 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
```

Se utilizzi Microsoft Windows 7 o Windows 10, puoi verificare se è possibile accedere a un endpoint dual-stack su IPv6 o IPv4 tramite il comando `ping`, come mostrato nell'esempio che segue.

```
ping ipv6.s3.dualstack.us-west-2.amazonaws.com
```

Utilizzo degli endpoint dual-stack Amazon S3

Gli endpoint dual-stack Amazon S3; supportano le richieste ai bucket S3 su IPv6 e IPv4. In questa sezione viene descritto come utilizzare gli endpoint dual-stack.

Argomenti

Endpoint dual-stack Amazon S3

Quando effettui una richiesta a un endpoint dual-stack, l'URL del bucket restituisce un indirizzo IPv6 o IPv4. Per ulteriori informazioni sull'accesso a un bucket su IPv6, consulta Esecuzione di richieste ad Amazon S3 su IPv6.

Quando si utilizza l'API REST, è possibile accedere direttamente a un endpoint Amazon S3 utilizzando il nome dell'endpoint (URI). È possibile accedere a un bucket S3 tramite un endpoint dual-stack utilizzando un nome di endpoint in stile hosting virtuale o in stile percorso. Amazon S3 supporta solo i nomi di endpoint dual-stack regionali, il che significa che è necessario specificare la regione come parte del nome.

Utilizzare le convenzioni di denominazione seguenti per i nomi di endpoint dual-stack in stile hosting virtuale e in stile percorso.

- Endpoint dual-stack in stile hosting virtuale:

  nomebucket.s3.dualstack.*regione-aws*.amazonaws.com

- Endpoint dual-stack in stile percorso

  s3.dualstack.*regione-aws*.amazonaws.com/*nomebucket*

Per ulteriori informazioni sullo stile del nome degli endpoint, consulta Accesso ed elenco di un bucket Amazon S3. Per un elenco di endpoint di Amazon S3, consultare la sezione relativa a regioni ed endpoint nella Riferimenti generali di AWS.

> **Important**
>
> È possibile utilizzare Transfer Acceleration con gli endpoint dual-stack. Per ulteriori informazioni, consulta Nozioni di base su Amazon S3 Transfer Acceleration.

> **Note**
>
> I due tipi di endpoint VPC per accedere ad Amazon S3 (gli endpoint VPC di interfaccia e gli endpoint VPC del gateway) non supportano il dual-stack. Per ulteriori informazioni sugli endpoint VPC per Amazon S3, consulta AWS PrivateLink per Amazon S3.

Quando usi AWS Command Line Interface (AWS CLI) e gli AWS SDK, puoi usare un parametro o un flag per passare a un endpoint dual-stack. È inoltre possibile specificare l'endpoint dual-stack direttamente come sostituzione dell'endpoint Amazon S3 nel file di configurazione. Le sezioni seguenti descrivono come utilizzare gli endpoint dual-stack degli e degli SDK. AWS CLI AWS

Utilizzo degli endpoint dual-stack di AWS CLI

Questa sezione fornisce esempi di AWS CLI comandi utilizzati per effettuare richieste a un endpoint dual-stack. Per istruzioni sulla configurazione di, vedere. AWS CLISviluppo con Amazon S3 tramite la AWS CLI

Imposta il valore `use_dualstack_endpoint` di configurazione su un profilo nel tuo AWS Config file per indirizzare tutte le richieste Amazon S3 effettuate dai `s3api` AWS CLI comandi `s3` and all'endpoint dual-stack per la regione specificata. `true` La regione va specificata nel file di configurazione o in un comando tramite l'opzione `--region`.

Quando si utilizzano endpoint dual-stack con, sono supportati entrambi gli stili di indirizzamento. AWS CLI`path` `virtual` Lo stile di indirizzamento, impostato nel file di configurazione, controlla se il nome del bucket si trova nel nome host o fa parte dell'URL. Per default, l'interfaccia a riga di comando (CLI) tenta di utilizzare lo stile virtuale, se possibile, tornando allo stile percorso, se necessario. Per ulteriori informazioni, consulta Configurazione di Amazon S3 per AWS CLI.

È anche possibile apportare modifiche alla configurazione tramite un comando, come mostrato nell'esempio seguente, dove `use_dualstack_endpoint` viene impostato su `true` e `addressing_style` viene impostato su `virtual` nel profilo di default.

```
$ aws configure set default.s3.use_dualstack_endpoint true
$ aws configure set default.s3.addressing_style virtual
```

Se desideri utilizzare un endpoint dual-stack solo per AWS CLI comandi specifici (non tutti i comandi), puoi utilizzare uno dei seguenti metodi:

- È possibile utilizzare l'endpoint dual-stack per comando, impostando il parametro `--endpoint-url` su `https://s3.dualstack.`*`aws-region`*`.amazonaws.com` o `http://s3.dualstack.`*`aws-region`*`.amazonaws.com` per un comando `s3` o `s3api`.

  ```
  $ aws s3api list-objects --bucket bucketname --endpoint-url https://s3.dualstack.aws-region.amazonaws.com
  ```

- Puoi impostare profili separati nel tuo file. AWS Config Ad esempio, si può creare un profilo che imposta `use_dualstack_endpoint` su `true` e un profilo che non imposta `use_dualstack_endpoint`. Quando si esegue un comando, specificare il profilo da usare, a seconda se si desidera o meno utilizzare l'endpoint dual-stack.

> **Note**
>
> Al momento non AWS CLI è possibile utilizzare l'accelerazione del trasferimento con endpoint dual-stack. Tuttavia, il supporto per il sarà presto disponibile. AWS CLI Per ulteriori informazioni, consulta Usando il AWS CLI.

Utilizzo degli endpoint dual-stack dagli SDK AWS

Questa sezione fornisce esempi di come accedere a un endpoint dual-stack utilizzando gli SDK. AWS

AWS SDK for Java esempio di endpoint dual-stack

Nell'esempio seguente viene mostrato come abilitare endpoint dual-stack durante la creazione di un client Amazon S3 utilizzando la AWS SDK for Java.

Per istruzioni su come creare e testare un esempio Java funzionante, consulta la Guida introduttiva alla Developer Guide. 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();
```

```
        }
    }
}
```

Se si utilizza AWS SDK for Java su Windows, potrebbe essere necessario impostare la seguente proprietà della macchina virtuale Java (JVM):

```
java.net.preferIPv6Addresses=true
```

AWS Esempio di endpoint dual-stack DI.NET SDK

Quando si utilizza l' AWS SDK for .NET, si utilizza `AmazonS3Config` la classe per abilitare l'uso di un endpoint dual-stack, come illustrato nell'esempio seguente.

```
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());
            }
        }
    }
}
```

Per un esempio .NET completo per l'elenco di oggetti, consulta Elenco delle chiavi oggetto a livello di programmazione.

Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

Utilizzo degli endpoint dual-stack dall'API REST

Per informazioni su come eseguire richieste su endpoint dual-stack tramite l'API REST, consulta Esecuzione di richieste a endpoint Dual-Stack utilizzando l'API REST.

Esecuzione di richieste tramite gli SDK AWS

Argomenti

È possibile inviare richieste autenticate ad Amazon S3 mediante SDK AWS o effettuando chiamate all'API REST direttamente nell'applicazione. L'API dell'SDK AWS utilizza le credenziali fornite per il calcolo della firma per l'autenticazione. Se si utilizza l'API REST direttamente nelle applicazioni, è necessario scrivere il codice necessario per calcolare la firma per l'autenticazione della richiesta. Per un elenco degli SDK AWS disponibili, consulta Codici di esempio e librerie.

Effettuare richieste utilizzando le Account AWS nostre credenziali utente IAM

Puoi utilizzare le tue credenziali di sicurezza Account AWS o quelle degli utenti IAM per inviare richieste autenticate ad Amazon S3. Questa sezione fornisce esempi di come inviare richieste autenticate utilizzando, e. AWS SDK for Java AWS SDK for .NET AWS SDK for PHP Per un elenco degli AWS SDK disponibili, vai a Codice di esempio e librerie.

Ciascuno di questi AWS SDK utilizza una catena di fornitori di credenziali specifica per l'SDK per trovare e utilizzare le credenziali ed eseguire azioni per conto del proprietario delle credenziali. Ciò che accomuna tutte queste catene di fornitori di credenziali è che tutte cercano il file di credenziali locale. AWS

Per ulteriori informazioni, consulta gli argomenti riportati di seguito.

Argomenti

Per creare un file di credenziali locale AWS

Il modo più semplice per configurare le credenziali per i tuoi AWS SDK consiste nell'utilizzare un AWS file di credenziali. Se usi il AWS Command Line Interface (AWS CLI), potresti aver già

configurato un file di AWS credenziali locale. Altrimenti, per impostare un file delle credenziali, attenersi alla procedura riportata di seguito.

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Creare un nuovo utente con autorizzazioni limitate ai servizi e alle operazioni a cui si desidera che il codice abbia accesso. Per ulteriori informazioni sulla creazione di un nuovo utente, consulta Creazione di utenti IAM (Console) e segui le istruzioni descritte fino al passaggio 8.
3. Scegliere Scarica .csv per salvare una copia locale delle credenziali AWS .
4. Sul computer, accedere alla directory principale e creare una directory `.aws`. Nei sistemi basati su Unix, ad esempio Linux oppure OS X, si trova nella seguente posizione:

   ```
   ~/.aws
   ```

   In Windows, si trova nella seguente posizione:

   ```
   %HOMEPATH%\.aws
   ```

5. Nella directory `.aws` creare un nuovo file denominato `credentials`.
6. Aprire il file delle credenziali `.csv` scaricato dalla console IAM e copiarne il contenuto nel file `credentials` utilizzando il seguente formato:

   ```
   [default]
   aws_access_key_id = your_access_key_id
   aws_secret_access_key = your_secret_access_key
   ```

7. Salvare il file `credentials` ed eliminare il file `.csv` scaricato al passaggio 3.

Il file delle credenziali condivise è ora configurato sul computer locale ed è pronto per essere utilizzato con gli AWS SDK.

Invio di richieste autenticate tramite gli SDK AWS

Utilizza gli AWS SDK per inviare richieste autenticate. Per ulteriori informazioni sull'invio di richieste autenticate, consulta Credenziali di sicurezza AWS o Autenticazione IAM Identity Center.

Java

Per inviare richieste autenticate ad Amazon S3 utilizzando le Account AWS tue credenziali utente o IAM, procedi come segue:

- Usare la classe `AmazonS3ClientBuilder` per creare una istanza `AmazonS3Client`.
- Eseguire uno dei metodi `AmazonS3Client` per inviare le richieste ad Amazon S3. Il client genera la firma necessaria dalle credenziali fornite e la include nella richiesta.

L'esempio seguente esegue le precedenti operazioni. Per informazioni sulla creazione e il test di un esempio funzionante, consulta Getting Started nella Developer Guide. 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

Per inviare richieste autenticate utilizzando le tue credenziali utente Account AWS o IAM:

- Crea un'istanza della classe `AmazonS3Client`.
- Eseguire uno dei metodi `AmazonS3Client` per inviare le richieste ad Amazon S3. Il client genera la firma necessaria dalle credenziali fornite e la include nella richiesta inviata ad Amazon S3.

Per ulteriori informazioni, consulta Effettuare richieste utilizzando le Account AWS nostre credenziali utente IAM.

> **Note**
>
> - È possibile creare il client `AmazonS3Client` senza specificare le credenziali di sicurezza. Le richieste inviate mediante questo client sono anonime e senza firma. Amazon S3 restituisce un errore se si inviano richieste anonime per una risorsa non disponibile pubblicamente.
> - Puoi creare un Account AWS e creare gli utenti richiesti. È inoltre possibile gestire le credenziali per tali utenti. Queste credenziali sono necessarie per eseguire l'attività nell'esempio seguente. Per ulteriori informazioni, consulta Configurazione delle credenziali di AWS nella Guida per gli sviluppatori di AWS SDK for .NET .
>
>   È quindi possibile anche configurare l'applicazione per recuperare attivamente profili e credenziali e quindi utilizzare esplicitamente tali credenziali durante la creazione di un client di servizio. AWS Per ulteriori informazioni, consulta Accesso a credenziali e profili in un'applicazione nella Guida per gli sviluppatori di AWS SDK for .NET .

L'esempio seguente di codice C# mostra come eseguire le precedenti operazioni. Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

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);
            }
        }
    }
}
```

Per gli esempi di utilizzo, consultare Panoramica degli oggetti di Amazon S3 e Panoramica dei bucket. Puoi testare questi esempi utilizzando le tue credenziali Account AWS o quelle di un utente IAM.

Ad esempio, per elencare tutte le chiavi degli oggetti incluse nel bucket, consultare Elenco delle chiavi oggetto a livello di programmazione.

PHP

Questa sezione spiega come utilizzare una classe della versione 3 di AWS SDK for PHP per inviare richieste autenticate utilizzando le tue credenziali utente Account AWS o IAM. Per ulteriori informazioni sull'API AWS SDK for Ruby, AWS vai a SDK for Ruby - Versione 2.

Il seguente esempio di codice PHP mostra come il client esegue una richiesta utilizzando le credenziali di sicurezza per elencare tutti i bucket per l'account.

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**
>
> È possibile creare il client `S3Client` senza specificare le credenziali di sicurezza. Le richieste inviate mediante questo client sono anonime e senza firma. Amazon S3 restituisce un errore se si inviano richieste anonime per una risorsa non disponibile pubblicamente. Per ulteriori informazioni, vedere Creazione di client anonimi nella Documentazione di AWS SDK for PHP.

Per esempi funzionanti, consulta Panoramica degli oggetti di Amazon S3. Puoi testare questi esempi utilizzando le tue credenziali utente Account AWS o quelle di IAM.

Per un esempio dell'elenco delle chiavi degli oggetti incluse in un bucket, consultare Elenco delle chiavi oggetto a livello di programmazione.

Ruby

Prima di poter utilizzare la versione 3 di AWS SDK for Ruby per effettuare chiamate ad Amazon S3, devi impostare le credenziali di AWS accesso utilizzate dall'SDK per verificare l'accesso ai tuoi bucket e oggetti. Se hai credenziali condivise configurate nel profilo delle AWS credenziali sul tuo sistema locale, la versione 3 dell'SDK for Ruby può utilizzare tali credenziali senza che tu debba dichiararle nel codice. Per ulteriori informazioni sull'impostazione di credenziali condivise, consulta Effettuare richieste utilizzando le Account AWS nostre credenziali utente IAM.

Il seguente frammento di codice Ruby utilizza le credenziali in un file di AWS credenziali condiviso su un computer locale per autenticare una richiesta per ottenere tutti i nomi delle chiavi degli oggetti in un bucket specifico. Esegue queste operazioni:

1. Crea un'istanza della classe `Aws::S3::Client`.
2. Effettua una richiesta ad Amazon S3 enumerando gli oggetti in un bucket utilizzando il metodo `list_objects_v2` di `Aws::S3::Client`. Il client genera il valore di firma necessario dalle credenziali nel file delle AWS credenziali sul tuo computer e lo include nella richiesta che invia ad Amazon S3.
3. Stampa la serie di nomi chiave degli oggetti al terminale.

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__
```

Se non disponi di un file di AWS credenziali locale, puoi comunque creare la `Aws::S3::Client` risorsa ed eseguire codice su bucket e oggetti Amazon S3. Le richieste che vengono inviate utilizzando la versione 3 di SDK per Ruby sono richieste anonime, senza firma, per impostazione predefinita. Amazon S3 restituisce un errore se si inviano richieste anonime per una risorsa non disponibile pubblicamente.

Puoi utilizzare ed espandere il precedente snippet di codice per applicazioni SDK per Ruby, come nel seguente esempio, più complesso.

```
# Prerequisites:
#  - An existing Amazon S3 bucket.

require "aws-sdk-s3"

# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if all operations succeed; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless list_bucket_objects?(s3_client, 'doc-example-bucket')
def list_bucket_objects?(s3_client, bucket_name)
  puts "Accessing the bucket named '#{bucket_name}'..."
  objects = s3_client.list_objects_v2(
    bucket: bucket_name,
```

```
    max_keys: 50
  )

  if objects.count.positive?
    puts "The object keys in this bucket are (first 50 objects):"
    objects.contents.each do |object|
      puts object.key
    end
  else
    puts "No objects found in this bucket."
  end

  return true
rescue StandardError => e
  puts "Error while accessing the bucket named '#{bucket_name}': #{e.message}"
  return false
end

# Example usage:
def run_me
  region = "us-west-2"
  bucket_name = "BUCKET_NAME"
  s3_client = Aws::S3::Client.new(region: region)

  exit 1 unless list_bucket_objects?(s3_client, bucket_name)
end

run_me if $PROGRAM_NAME == __FILE__
```

Go

Example

L'esempio seguente utilizza AWS le credenziali caricate automaticamente dall'SDK for Go dal file delle credenziali condiviso.

```
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)
  }
 }
}
```

Risorse correlate

- Sviluppo con Amazon S3 utilizzando gli SDK AWS
- AWS SDK for PHP per Amazon S3 Aws\S3\S3Client Class
- Documentazione AWS SDK for PHP

Esecuzione di richieste mediante le credenziali temporanee per gli utenti IAM

Un utente IAM Account AWS o un utente IAM può richiedere credenziali di sicurezza temporanee e utilizzarle per inviare richieste autenticate ad Amazon S3. In questa sezione sono riportati esempi sull'utilizzo degli SDK AWS SDK for Java, .NET e PHP per ottenere credenziali di sicurezza temporanee con cui autenticare le richieste ad Amazon S3.

Java

Un utente IAM o un utente Account AWS può richiedere credenziali di sicurezza temporanee (vediEsecuzione di richieste) utilizzando AWS SDK for Java e utilizzarle per accedere ad Amazon S3. Queste credenziali scadono al termine della durata della sessione specificata.

Per default, la sessione dura un'ora. Se si utilizzano le credenziali utente IAM, è possibile specificare la durata della richiesta delle credenziali di protezione temporanee da 15 minuti alla durata massima della sessione per il ruolo. Per ulteriori informazioni sulle credenziali di sicurezza temporanee, consulta la sezione relativa alle credenziali di sicurezza temporanee nella Guida per l'utente IAM. Per ulteriori informazioni sull'esecuzione di richieste, consulta Esecuzione di richieste.

Per ottenere credenziali di sicurezza temporanee e accedere ad Amazon S3

1. Crea un'istanza della classe `AWSSecurityTokenService`. Per ulteriori informazioni sulla specifica delle credenziali, consulta Sviluppo con Amazon S3 utilizzando gli SDK AWS.
2. Recuperare le credenziali di sicurezza temporanee per il ruolo desiderato chiamando il metodo `assumeRole()` del client Security Token Service (STS).
3. Creare un pacchetto di credenziali di sicurezza temporanee in un oggetto `BasicSessionCredentials`. Tale oggetto viene utilizzato per specificare le credenziali di sicurezza temporanee per il client Amazon S3.
4. Creare un'istanza della classe `AmazonS3Client` utilizzando le credenziali di sicurezza temporanee. Le richieste vengono inviate ad Amazon S3 con questo client. In caso di invio di richieste con credenziali scadute, Amazon S3 restituirà un errore.

> **Note**
>
> Se si ottengono credenziali di sicurezza temporanee utilizzando le credenziali di sicurezza dell' Account AWS , le credenziali temporanee rimarranno valide solo per un'ora. È

possibile specificare la durata della sessione solo se si utilizzano le credenziali utente Amazon S3 per richiedere una sessione.

L'esempio seguente elenca un set di chiavi degli oggetti incluse nel bucket specificato. L'esempio ottiene le credenziali di sicurezza temporanee per una sessione e le utilizza per inviare una richiesta autenticata ad Amazon S3.

Per testare l'esempio utilizzando le credenziali di un utente IAM, è necessario creare un utente IAM nell' Account AWS. Per ulteriori informazioni su come creare un utente IAM, consulta Creazione del primo utente IAM e del gruppo di amministratori nella Guida per l'utente di IAM.

Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
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 utente IAM o un utente Account AWS può richiedere credenziali di sicurezza temporanee utilizzando AWS SDK for .NET e utilizzarle per accedere ad Amazon S3. Queste credenziali scadono al termine della durata della sessione.

Per default, la sessione dura un'ora. Se si utilizzano le credenziali utente IAM, è possibile specificare la durata della richiesta delle credenziali di protezione temporanee da 15 minuti alla durata massima della sessione per il ruolo. Per ulteriori informazioni sulle credenziali di sicurezza temporanee, consulta la sezione relativa alle credenziali di sicurezza temporanee nella Guida per l'utente IAM. Per ulteriori informazioni sull'esecuzione di richieste, consulta Esecuzione di richieste.

Per ottenere credenziali di sicurezza temporanee e accedere ad Amazon S3

1. Crea un'istanza del AWS Security Token Service client,. `AmazonSecurityTokenServiceClient` Per ulteriori informazioni sulla specifica delle credenziali, consulta Sviluppo con Amazon S3 utilizzando gli SDK AWS.
2. Avviare una sezione richiamando il metodo `GetSessionToken` del client STS creato nella fase precedente. Le informazioni sulla sessione vengono fornite al metodo tramite un oggetto `GetSessionTokenRequest`.

   Il metodo restituisce le credenziali di sicurezza temporanee.
3. Creare un pacchetto di credenziali di sicurezza temporanee in un'istanza dell'oggetto `SessionAWSCredentials`. Tale oggetto viene utilizzato per specificare le credenziali di sicurezza temporanee per il client Amazon S3.
4. Creare un'istanza della classe `AmazonS3Client` passando le credenziali di sicurezza temporanee. Le richieste vengono inviate ad Amazon S3 con questo client. In caso di invio di richieste mediante l'utilizzo di credenziali scadute, Amazon S3 restituisce un errore.

> **Note**
>
> Se si ottengono credenziali di sicurezza temporanee utilizzando le credenziali di sicurezza dell' Account AWS , le credenziali temporanee rimarranno valide solo per un'ora.
> Puoi specificare la durata di una sessione solo se utilizzi le credenziali utente IAM per richiedere una sessione.

L'esempio di codice C# seguente elenca le chiavi degli oggetti incluse nel bucket specificato. A titolo illustrativo, l'esempio ottiene le credenziali di sicurezza temporanee per una sessione di default della durata di un'ora e le utilizza per inviare una richiesta autenticata ad Amazon S3.

Per testare l'esempio utilizzando le credenziali di un utente IAM, è necessario creare un utente IAM nell' Account AWS. Per ulteriori informazioni su come creare un utente IAM, consulta Creazione del primo utente IAM e del gruppo di amministratori nella Guida per l'utente di IAM. Per ulteriori informazioni sull'esecuzione di richieste, consulta Esecuzione di richieste.

Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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

Per ulteriori informazioni sull'API AWS SDK for Ruby, AWS vai a SDK for Ruby - Versione 2.

Un utente IAM o un utente Account AWS può richiedere credenziali di sicurezza temporanee utilizzando la versione 3 di. AWS SDK for PHP Può quindi utilizzare tali credenziali per accedere ad Amazon S3. Le credenziali scadono alla fine della durata della sessione.

Per default, la sessione dura un'ora. Se si utilizzano le credenziali utente IAM, è possibile specificare la durata della richiesta delle credenziali di protezione temporanee da 15 minuti alla durata massima della sessione per il ruolo. Per ulteriori informazioni sulle credenziali di sicurezza temporanee, consulta la sezione relativa alle credenziali di sicurezza temporanee nella Guida per l'utente IAM. Per ulteriori informazioni sull'esecuzione di richieste, consulta Esecuzione di richieste.

> **Note**
>
> Se si ottengono credenziali di sicurezza temporanee utilizzando le credenziali di sicurezza dell' Account AWS , le credenziali di sicurezza temporanee rimarranno valide solo per

un'ora. È possibile specificare la durata della sessione solo se si utilizzano le credenziali utente Amazon S3 per richiedere una sessione.

Example

L'esempio di codice PHP seguente elenca le chiavi degli oggetti incluse nel bucket specificato utilizzando credenziali di sicurezza temporanee. L'esempio ottiene le credenziali di sicurezza temporanee per una sessione di default della durata di un'ora e le utilizza per inviare una richiesta autenticata ad Amazon S3. Per ulteriori informazioni sull'API AWS SDK for Ruby, AWS vai a SDK for Ruby - Versione 2.

Per testare l'esempio utilizzando le credenziali di un utente IAM, è necessario creare un utente IAM nell' Account AWS. Per informazioni su come creare un utente IAM, consulta Creazione del primo utente IAM e del gruppo di amministratori nella Guida per l'utente IAM. Per degli esempi di impostazione della durata della sessione quando si utilizzano le credenziali utente IAM per richiedere una sessione, consulta Esecuzione di richieste mediante le credenziali temporanee per gli utenti 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 utente IAM o un utente Account AWS può richiedere credenziali di sicurezza AWS SDK for Ruby temporanee utilizzandole per accedere ad Amazon S3. Queste credenziali scadono al termine della durata della sessione.

Per default, la sessione dura un'ora. Se si utilizzano le credenziali utente IAM, è possibile specificare la durata della richiesta delle credenziali di protezione temporanee da 15 minuti alla durata massima della sessione per il ruolo. Per ulteriori informazioni sulle credenziali di sicurezza temporanee, consulta la sezione relativa alle credenziali di sicurezza temporanee nella Guida per l'utente IAM. Per ulteriori informazioni sull'esecuzione di richieste, consulta Esecuzione di richieste.

> **Note**
>
> Se si ottengono credenziali di sicurezza temporanee utilizzando le credenziali di sicurezza dell' Account AWS , le credenziali di sicurezza temporanee rimarranno valide solo per un'ora. Puoi specificare la durata della sessione solo se utilizzi le credenziali utente IAM per richiedere una sessione.

Nel seguente esempio di codice Ruby viene creato un utente temporaneo per elencare le voci in un bucket specificato per un'ora. Per utilizzare questo esempio, devi disporre di AWS credenziali con le autorizzazioni necessarie per creare nuovi AWS Security Token Service (AWS STS) client ed elencare i bucket 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
```

Risorse correlate

- Sviluppo con Amazon S3 utilizzando gli SDK AWS
- AWS SDK for PHP per la classe Amazon S3 Aws\ S3\ S3Client
- Documentazione AWS SDK for PHP

Esecuzione di richieste mediante le credenziali temporanee per gli utenti federati

Puoi richiedere credenziali di sicurezza temporanee e fornirle agli utenti o alle applicazioni federati che devono accedere alle tue AWS risorse. Questa sezione fornisce esempi di come utilizzare l' AWS SDK per ottenere credenziali di sicurezza temporanee per utenti o applicazioni federati e inviare richieste autenticate ad Amazon S3 utilizzando tali credenziali. Per un elenco degli AWS SDK disponibili, consulta Codice di esempio e librerie.

> **Note**
>
> Account AWS Sia l'utente che un utente IAM possono richiedere credenziali di sicurezza temporanee per gli utenti federati. Tuttavia, per una maggiore sicurezza, la richiesta di credenziali temporanee è consentita solo a un utente IAM che dispone delle autorizzazioni necessarie, in modo da garantire che l'utente federato riceva esclusivamente le autorizzazioni dell'utente IAM che presenta la richiesta. In alcune applicazioni, può risultare utile creare un utente IAM con specifiche autorizzazioni finalizzate a concedere credenziali di accesso temporanee agli utenti federati e alle applicazioni.

Java

Puoi fornire credenziali di sicurezza temporanee per gli utenti e le applicazioni federati in modo che possano inviare richieste autenticate per accedere alle tue risorse. AWS Quando si richiedono le credenziali temporanee, è necessario fornire un nome utente e una policy IAM che descrive le autorizzazioni a livello di risorsa che si desidera concedere. Per default, la sessione dura un'ora. In fase di richiesta delle credenziali di sicurezza temporanee per utenti federati e applicazioni, è possibile impostare in modo esplicito un valore di durata diverso.

> **Note**
>
> Per una maggiore sicurezza nella richiesta di credenziali di sicurezza temporanee per utenti federati e applicazioni, consigliamo di utilizzare un utente IAM dedicato che abbia solo le autorizzazioni di accesso necessarie. L'utente temporaneo che si crea non può mai avere autorizzazioni più ampie rispetto all'utente IAM che ha richiesto le credenziali di sicurezza temporanee. Per ulteriori informazioni, consulta FAQ AWS Identity and Access Management.

Per fornire credenziali di sicurezza e inviare una richiesta autenticata per accedere alle risorse, procedere come segue:

- Crea un'istanza della classe `AWSSecurityTokenServiceClient`.
- Avviare una sezione richiamando il metodo `getFederationToken()` del client Security Token Service (STS). Fornisci informazioni sulla sessione, incluso il nome utente e la policy IAM che si desidera collegare alle credenziali temporanee. È possibile specificare una durata della sessione opzionale. Questo metodo restituisce le credenziali di sicurezza temporanee.
- Creare un pacchetto di credenziali di sicurezza temporanee in un'istanza dell'oggetto `BasicSessionCredentials`. Tale oggetto viene utilizzato per specificare le credenziali di sicurezza temporanee per il client Amazon S3.
- Creare un'istanza della classe `AmazonS3Client` utilizzando le credenziali di sicurezza temporanee. Le richieste vengono inviate ad Amazon S3 con questo client. In caso di invio di richieste mediante l'utilizzo di credenziali scadute, Amazon S3 restituisce un errore.

Example

L'esempio elenca le chiavi nel bucket S3 specificato. Nell'esempio, è necessario ottenere le credenziali di sicurezza temporanee per l'utente federato per una sessione della durata di due ore e utilizzare le credenziali per inviare richieste autenticate ad Amazon S3. Per eseguire l'esempio, devi creare un utente IAM con una policy allegata che consenta all'utente di richiedere credenziali di sicurezza temporanee ed elencare le tue risorse. AWS Questa policy esegue quanto segue:

```
{
  "Statement":[{
      "Action":["s3:ListBucket",
        "sts:GetFederationToken*"
      ],
      "Effect":"Allow",
      "Resource":"*"
    }
  ]
}
```

Per ulteriori informazioni su come creare un utente IAM, consulta Creazione del primo utente IAM e del gruppo di amministratori nella Guida per l'utente di IAM.

Dopo la creazione dell'utente IAM e il collegamento della policy di cui sopra, è possibile eseguire l'esempio seguente. Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella AWS SDK for Java Developer Guide.

```
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

Puoi fornire credenziali di sicurezza temporanee per gli utenti e le applicazioni federati in modo che possano inviare richieste autenticate per accedere alle tue risorse. AWS Quando si richiedono le credenziali temporanee, è necessario fornire un nome utente e una policy IAM che descrive le autorizzazioni a livello di risorsa che si desidera concedere. Per impostazione predefinita, la sessione dura un'ora. In fase di richiesta delle credenziali di sicurezza temporanee per utenti federati e applicazioni, è possibile impostare in modo esplicito un valore di durata diverso. Per informazioni sull'invio di richieste autenticate, consulta Esecuzione di richieste.

> **Note**
>
> Nel richiedere le credenziali di sicurezza temporanee per utenti federati e applicazioni, per una maggiore sicurezza, consigliamo di utilizzare un utente IAM dedicato che abbia solo le autorizzazioni di accesso necessarie. L'utente temporaneo che si crea non può mai avere autorizzazioni più ampie rispetto all'utente IAM che ha richiesto le credenziali di sicurezza temporanee. Per ulteriori informazioni, consulta FAQ AWS Identity and Access Management.

Esegui le operazioni indicate di seguito:

- Crea un'istanza del AWS Security Token Service client, classe. `AmazonSecurityTokenServiceClient`
- Avviare una sessione chiamando il metodo `GetFederationToken` del client STS. Occorre fornire informazioni sulla sessione, incluso il nome utente e la policy IAM da collegare alle credenziali temporanee. Eventualmente, è possibile specificare una durata della sessione. Questo metodo restituisce le credenziali di sicurezza temporanee.
- Creare un pacchetto di credenziali di sicurezza temporanee in un'istanza dell'oggetto `SessionAWSCredentials`. Tale oggetto viene utilizzato per specificare le credenziali di sicurezza temporanee per il client Amazon S3.
- Creare un'istanza della classe `AmazonS3Client` passando le credenziali di sicurezza temporanee. Con questo client, le richieste vengono inviate ad Amazon S3. In caso di invio di richieste mediante l'utilizzo di credenziali scadute, Amazon S3 restituisce un errore.

Example

L'esempio C# seguente elenca le chiavi incluse nel bucket specificato. Nell'esempio, è necessario ottenere le credenziali di sicurezza temporanee per l'utente federato (User1) per una sessione della durata di due ore e utilizzare le credenziali per inviare richieste autenticate a Amazon S3.

- Per questo esercizio, si crea un utente IAM con le autorizzazioni minime. Utilizzando le credenziali di questo utente IAM, si richiedono le credenziali temporanee per gli altri. In questo esempio sono elencati solo gli oggetti di un bucket specifico. Crea un utente IAM con la seguente policy collegata:

```
{
  "Statement":[{
      "Action":["s3:ListBucket",
        "sts:GetFederationToken*"
      ],
      "Effect":"Allow",
      "Resource":"*"
    }
  ]
}
```

  La policy consente all'utente IAM di richiedere credenziali di sicurezza temporanee e autorizzazioni di accesso solo per elencare AWS le proprie risorse. Per ulteriori informazioni su come creare un utente IAM, consulta Creazione del primo utente IAM e del gruppo di amministratori nella Guida per l'utente di IAM.

- Utilizza le credenziali di sicurezza dell'utente IAM per provare l'esempio seguente. Nell'esempio vengono inviate richieste autenticate ad Amazon S3 mediante le credenziali di sicurezza temporanee. Nell'esempio è specificata la seguente policy durante la richiesta di credenziali di sicurezza temporanee per l'utente federato (User1) con accesso limitato agli elenchi di oggetti di un bucket specifico (`YourBucketName`). Occorre specificare la policy e fornire il nome del bucket esistente.

```
{
  "Statement":[
    {
      "Sid":"1",
      "Action":["s3:ListBucket"],
      "Effect":"Allow",
      "Resource":"arn:aws:s3:::YourBucketName"
```

```
    }
  ]
}
```

- Example

  Aggiorna il seguente esempio e fornire il nome del bucket specificato nella precedente policy di accesso dell'utente federato. Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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

Questo argomento spiega come utilizzare le classi della versione 3 AWS SDK for PHP per richiedere credenziali di sicurezza temporanee per utenti e applicazioni federati e utilizzarle per accedere alle risorse archiviate in Amazon S3. Per ulteriori informazioni sull'API AWS SDK for Ruby, AWS vai a SDK for Ruby - Versione 2.

Puoi fornire credenziali di sicurezza temporanee agli utenti e alle applicazioni federati in modo che possano inviare richieste autenticate per accedere alle tue risorse. AWS Quando si richiedono le credenziali temporanee, è necessario fornire un nome utente e una policy IAM che descrive le autorizzazioni a livello di risorsa che si desidera concedere. Queste credenziali scadono alla scadenza della durata della sessione. Per default, la sessione dura un'ora. In fase di richiesta delle credenziali di sicurezza temporanee per utenti federati e applicazioni, è possibile impostare

in modo esplicito un valore di durata diverso. Per ulteriori informazioni sulle credenziali di sicurezza temporanee, consulta la sezione relativa alle credenziali di sicurezza temporanee nella Guida per l'utente IAM. Per ulteriori informazioni sulla specifica delle credenziali di sicurezza temporanee alle applicazioni e agli utenti federati, consulta Esecuzione di richieste.

Per una maggiore sicurezza nella richiesta di credenziali di sicurezza temporanee per utenti federati e applicazioni, consigliamo di utilizzare un utente IAM dedicato che abbia solo le autorizzazioni di accesso necessarie. L'utente temporaneo che si crea non può mai avere autorizzazioni più ampie rispetto all'utente IAM che ha richiesto le credenziali di sicurezza temporanee. Per ulteriori informazioni, consulta l'articolo sulla federazione delle identità Web in Domande frequenti di AWS Identity and Access Management.

Per ulteriori informazioni sull'API AWS SDK for Ruby, AWS vai a SDK for Ruby - Versione 2.

Example

L'esempio PHP seguente elenca le chiavi incluse nel bucket specificato. Nell'esempio, è necessario ottenere le credenziali di sicurezza temporanee per l'utente federato (User1) per una sessione della durata di un'ora. Poi si utilizzano le credenziali di sicurezza temporanee per inviare richieste autenticate ad Amazon S3.

Per una maggiore sicurezza quando si effettua la richiesta di credenziali temporanee per altri soggetti, si utilizzano le credenziali di sicurezza di un utente IAM che abbia le autorizzazioni per richiedere credenziali di sicurezza temporanee. Per concedere all'utente IAM solo le autorizzazioni minime specifiche delle applicazioni per l'utente federato, puoi limitare anche le autorizzazioni di accesso dell'utente IAM. In questo esempio sono elencati solo gli oggetti di un bucket specifico. Crea un utente IAM con la seguente policy collegata:

```
{
  "Statement":[{
      "Action":["s3:ListBucket",
        "sts:GetFederationToken*"
      ],
      "Effect":"Allow",
      "Resource":"*"
    }
  ]
}
```

La policy consente all'utente IAM di richiedere credenziali di sicurezza temporanee e autorizzazioni di accesso solo per elencare le proprie risorse. AWS Per ulteriori informazioni

su come creare un utente IAM, consulta Creazione del primo utente IAM e del gruppo di amministratori nella Guida per l'utente di IAM.

Puoi ora utilizzare le credenziali di sicurezza dell'utente IAM per provare l'esempio seguente. Nell'esempio viene inviata una richiesta autenticata ad Amazon S3 mediante credenziali di sicurezza temporanee. Durante la richiesta di credenziali di sicurezza temporanee per l'utente federato (User1), nell'esempio è specificata la seguente policy, con accesso limitato agli oggetti dell'elenco di un bucket specifico. Aggiornare la policy con il nome del bucket.

```
{
  "Statement":[
    {
      "Sid":"1",
      "Action":["s3:ListBucket"],
      "Effect":"Allow",
      "Resource":"arn:aws:s3:::YourBucketName"
    }
  ]
}
```

Nell'esempio che segue, quando specifichi la risorsa della policy, è necessario sostituire `YourBucketName` con il nome del tuo 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

Puoi fornire credenziali di sicurezza temporanee agli utenti e alle applicazioni federati in modo che possano inviare richieste autenticate per accedere alle tue risorse. AWS Quando si richiedono tali credenziali temporanee a un servizio IAM, è necessario fornire un nome utente e una policy IAM che descrive le autorizzazioni a livello di risorsa che si desidera concedere. Per default, la sessione dura un'ora. Tuttavia, se si sta effettuando la richiesta di credenziali temporanee mediante credenziali utente IAM, è possibile definire espressamente un valore diverso per la durata quando si richiedono credenziali di sicurezza temporanee per utenti federati e applicazioni.

Per ulteriori informazioni sulle credenziali di sicurezza temporanee per le applicazioni e gli utenti federati, consulta Esecuzione di richieste.

> **Note**
>
> Per una maggiore sicurezza, quando richiedi credenziali di sicurezza temporanee per utenti federati e applicazioni, potrebbe essere preferibile usare un utente IAM dedicato che abbia solo le autorizzazioni di accesso necessarie. L'utente temporaneo che si crea non può mai avere autorizzazioni più ampie rispetto all'utente IAM che ha richiesto le credenziali di sicurezza temporanee. Per ulteriori informazioni, consulta FAQ AWS Identity and Access Management.

Example

Nel seguente esempio di codice Ruby consente a un utente con una serie limitata di autorizzazione per visualizzare le chiavi nel bucket specificato.

```
# 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__
```

Risorse correlate

- Sviluppo con Amazon S3 utilizzando gli SDK AWS
- AWS SDK for PHP per la classe Amazon S3 Aws\ S3\ S3Client
- Documentazione AWS SDK for PHP

Esecuzione di richieste con l'utilizzo di API REST

Questa sezione contiene informazioni su come effettuare richieste agli endpoint Amazon S3 utilizzando l'API REST. Per un elenco di endpoint di Amazon S3, consultare la sezione relativa a regioni ed endpoint nella Riferimenti generali di AWS.

Costruzione di nomi host S3 per richieste API REST

Gli endpoint Amazon S3 seguono la struttura riportata di seguito:

```
s3.Region.amazonaws.com
```

Gli endpoint degli Punti di accesso Amazon S3 e gli endpoint dual-stack seguono la struttura standard:

- Access point Amazon S -`s3-accesspoint.`*`Region`*`.amazonaws.com`
- Dual-stack - `s3.dualstack.`*`Region`*`.amazonaws.com`

Per un elenco completo degli endpoint e delle regioni Amazon S3, consultare la sezione relativa a endpoint e quote di Amazon S3 nella Riferimenti generali di Amazon Web Services.

Richieste in stile hosting virtuale e in stile percorso

Quando si effettua una richiesta utilizzando l'API REST, è possibile utilizzare URI in stile hosting virtuale o in stile percorso per gli endpoint Amazon S3. Per ulteriori informazioni, consulta Hosting virtuale dei bucket.

Example Richiesta in stile hosting virtuale

Qui di seguito è riportato un esempio di richiesta in stile hosting virtuale per eliminare il file `puppy.jpg` dal bucket denominato `examplebucket` nella regione Stati Uniti occidentali (Oregon). Per ulteriori informazioni sulle richieste in stile hosting virtuale, consulta Richieste in stile hosting virtuale.

```
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 Richiesta in stile percorso

Di seguito è riportato un esempio di una versione in stile percorso della stessa richiesta.

```
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
```

Attualmente, Amazon S3 supporta sia gli URL in stile hosting virtuale che quelli in stile percorso in tutte le Regioni AWS. Tuttavia, gli URL in stile percorso non saranno più disponibili in futuro. Per ulteriori informazioni, consulta la seguente nota importante.

Per ulteriori informazioni sulle richieste in stile percorso, consulta Richieste in stile percorso.

> **Important**
>
> Aggiornamento (23 settembre 2020): per assicurare che i clienti abbiano il tempo necessario per eseguire la transizione agli URL in stile hosting virtuale, abbiamo deciso di posticipare l'obsolescenza degli URL in stile percorso. Per ulteriori informazioni, consulta Piano di obsolescenza del percorso Amazon S3 - Il resto della storia nel Blog AWS News.

Esecuzione di richieste a endpoint Dual-Stack utilizzando l'API REST

Quando si utilizza l'API REST, è possibile accedere direttamente a un endpoint dual-stack utilizzando un nome di endpoint (URI) in stile hosting virtuale o in stile percorso. Tutti i nomi di endpoint dual-stack Amazon S3 includono la regione nel nome. A differenza degli endpoint solo IPv4 standard, sia gli endpoint in stile hosting virtuale sia quelli in stile percorso utilizzano nomi di endpoint specifici per regione.

Example Richiesta con endpoint dual-stack in stile hosting virtuale

È possibile utilizzare un endpoint in stile hosting virtuale nella richiesta REST come illustrato nell'esempio seguente che recupera l'oggetto `puppy.jpg` dal bucket denominato `examplebucket` nella regione Stati Uniti occidentali (Oregon).

```
GET /puppy.jpg HTTP/1.1
Host: examplebucket.s3.dualstack.us-west-2.amazonaws.com
Date: Mon, 11 Apr 2016 12:00:00 GMT
x-amz-date: Mon, 11 Apr 2016 12:00:00 GMT
Authorization: authorization string
```

Example Richiesta con endpoint dual-stack in stile percorso

Altrimenti, è possibile utilizzare un endpoint in stile percorso nella richiesta come illustrato nell'esempio seguente.

```
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
```

Per ulteriori informazioni sugli endpoint dual-stack, consulta Utilizzo degli endpoint dual-stack Amazon S3.

Per ulteriori informazioni su come effettuare richieste utilizzando l'API REST, consulta gli argomenti di seguito.

Argomenti

- Hosting virtuale dei bucket
- Reindirizzamento delle richieste e API REST

Hosting virtuale dei bucket

L'hosting virtuale è la pratica di servire più siti Web da un unico server Web. Un modo per differenziare i siti nelle richieste REST API di Amazon S3 consiste nell'utilizzare il nome host apparente dell'URI della richiesta anziché semplicemente la parte del nome del percorso dell'URI. Una richiesta REST Amazon S3 ordinaria specifica un bucket utilizzando il primo componente delimitato da barre del percorso dell'URI della richiesta. È possibile invece utilizzare l'hosting virtuale di Amazon S3 per un bucket in una chiamata di REST API utilizzando l'intestazione HTTP `Host`. In pratica, Amazon S3 interpreta `Host` come se la maggior parte dei bucket fosse accessibile immediatamente per alcuni tipi di richieste all'indirizzo `https://bucket-name.s3.region-code.amazonaws.com`. Per un elenco completo degli endpoint e delle regioni Amazon S3, consultare la sezione relativa a endpoint e quote di Amazon S3 nella Riferimenti generali di Amazon Web Services.

L'hosting virtuale ha anche altri vantaggi. Assegnando al bucket il nome del dominio registrato e rendendo tale nome un alias DNS per Amazon S3, è possibile personalizzare completamente l'URL delle risorse Amazon S3, ad esempio, `http://my.bucket-name.com/`. Puoi anche pubblicare nella directory root del server virtuale del bucket. Questa possibilità può essere importante in quanto molte applicazioni esistenti cercano i file in questa ubicazione standard. Ad esempio, `favicon.ico`, `robots.txt` e `crossdomain.xml` dovrebbero trovarsi tutti nella directory root.

> **Important**
>
> Quando si utilizzano bucket in stile hosting virtuale con SSL, il certificato jolly SSL confronta solo i bucket che non contengono punti (`.`). Per risolvere questa limitazione, utilizzare HTTP

o scrivere una logica di verifica del certificato personalizzata. Per ulteriori informazioni, consulta Amazon S3 Path Deprecation Plan (Piano di obsolescenza del percorso Amazon S3) in AWS News Blog.

Argomenti

Richieste in stile percorso

Attualmente, Amazon S3 supporta sia gli URL in stile hosting virtuale che quelli in stile percorso in tutte le Regioni AWS. Tuttavia, gli URL in stile percorso non saranno più disponibili in futuro. Per ulteriori informazioni, consulta la seguente nota importante.

In Amazon S3, gli URL in stile percorso utilizzano il seguente formato.

```
https://s3.region-code.amazonaws.com/bucket-name/key-name
```

Ad esempio, se hai creato un bucket denominato `DOC-EXAMPLE-BUCKET1` nella regione Stati Uniti occidentali (Oregon) e vuoi accedere all'oggetto `puppy.jpg` in quel bucket, puoi utilizzare il seguente URL in stile percorso:

```
https://s3.us-west-2.amazonaws.com/DOC-EXAMPLE-BUCKET1/puppy.jpg
```

Important

Aggiornamento (23 settembre 2020): per assicurare che i clienti abbiano il tempo necessario per eseguire la transizione agli URL in stile hosting virtuale, abbiamo deciso di posticipare

l'obsolescenza degli URL in stile percorso. Per ulteriori informazioni, consulta Piano di obsolescenza del percorso Amazon S3 - Il resto della storia nel Blog AWS News.

> **Warning**
>
> In caso di hosting di contenuti di siti Web a cui sarà possibile accedere da un browser Web, evita di utilizzare URL in stile percorso, che potrebbero interferire con il modello di sicurezza di origine del browser. Per l'hosting dei contenuti dei siti Web, ti consigliamo di utilizzare gli endpoint del sito Web S3 o una distribuzione CloudFront. Per ulteriori informazioni, consulta Endpoint del sito Web e l'argomento relativo all'implementazione di un'applicazione a pagina singola basata su React in Amazon S3 e CloudFront nel manuale AWS Perspective Guidance Patterns.

Richieste in stile hosting virtuale

In un URI in stile hosting virtuale, il nome del bucket fa parte del nome del dominio nell'URL.

Gli URL in stile hosting virtuale di Amazon S3 utilizzano il seguente formato.

```
https://bucket-name.s3.region-code.amazonaws.com/key-name
```

In questo esempio, `DOC-EXAMPLE-BUCKET1` è il nome del bucket, Stati Uniti occidentali (Oregon) è la regione e `puppy.png` è il nome della chiave:

```
https://DOC-EXAMPLE-BUCKET1.s3.us-west-2.amazonaws.com/puppy.png
```

Specifica del bucket nell'intestazione HTTP **`Host`**

Purché la richiesta `GET` non utilizzi l'endpoint SSL, è possibile specificare il bucket per la richiesta utilizzando l'intestazione HTTP `Host`. L'intestazione `Host` in una richiesta REST viene interpretata come indicato di seguito:

- Se l'intestazione `Host` viene omessa o ha un valore `s3.region-code.amazonaws.com`, il bucket per la richiesta sarà il primo componente delimitato da barre del percorso dell'URI della richiesta e la chiave per la richiesta sarà composta dai restanti componenti dell'URI della richiesta. Questo è il metodo ordinario, mostrato nel primo e nel secondo esempio di questa sezione. È possibile omettere l'intestazione `Host` solo per le richieste HTTP 1.0.

- Altrimenti, se il valore dell'intestazione `Host` termina con `.s3.`*`region-code`*`.amazonaws.com`, il nome del bucket è il componente principale del valore dell'intestazione `Host` fino a `.s3.`*`region-code`*`.amazonaws.com`. La chiave per la richiesta è l'URI della richiesta. Questa interpretazione espone i bucket come sottodomini di `.s3.`*`region-code`*`.amazonaws.com`, come mostrano il terzo e il quarto esempio in questa sezione.
- Altrimenti, il bucket per la richiesta è il valore in caratteri minuscoli dell'intestazione `Host` e la chiave per la richiesta è l'URI della richiesta. Questa interpretazione è utile nei casi in cui il nome DNS è stato registrato come nome del bucket e configurato il nome come alias del nome canonico (CNAME) per Amazon S3. La procedura per registrare i nomi di dominio e configurare i record DNS CNAME esula dall'ambito di questa guida, ma il risultato è mostrato nell'esempio finale di questa sezione.

Esempi

Questa sezione fornisce URL e richieste di esempio.

Example - URL e richieste in stile percorso

In questo esempio viene utilizzato:

- Nome bucket - `example.com`
- Regione - Stati Uniti orientali (Virginia settentrionale)
- Nome chiave - `homepage.html`

Di seguito è riportato l'URL:

```
http://s3.us-east-1.amazonaws.com/example.com/homepage.html
```

Di seguito è riportata la richiesta:

```
GET /example.com/homepage.html HTTP/1.1
Host: s3.us-east-1.amazonaws.com
```

Di seguito è riportata la richiesta con HTTP 1.0 e senza intestazione `Host`:

```
GET /example.com/homepage.html HTTP/1.0
```

Per informazioni sui nomi compatibili con DNS, consulta Limitazioni. Per ulteriori informazioni sulle chiavi, consultare Chiavi.

Example - URL e richieste in stile hosting virtuale

In questo esempio viene utilizzato:

- Nome bucket - `DOC-EXAMPLE-BUCKET1`
- Regione - Europa (Irlanda)
- Nome chiave - `homepage.html`

Di seguito è riportato l'URL:

```
http://DOC-EXAMPLE-BUCKET1.s3.eu-west-1.amazonaws.com/homepage.html
```

Di seguito è riportata la richiesta:

```
GET /homepage.html HTTP/1.1
Host: DOC-EXAMPLE-BUCKET1.s3.eu-west-1.amazonaws.com
```

Example - metodo alias CNAME

Per utilizzare questo metodo, è necessario configurare il nome DNS come un alias CNAME per *`bucket-name`*`.s3.us-east-1.amazonaws.com`. Per ulteriori informazioni, consulta Personalizzazione degli URL Amazon S3 con record CNAME.

In questo esempio viene utilizzato:

- Nome bucket - `example.com`
- Nome chiave - `homepage.html`

Di seguito è riportato l'URL:

```
http://www.example.com/homepage.html
```

Di seguito è riportato l'esempio:

```
GET /homepage.html HTTP/1.1
```

```
Host: www.example.com
```

Personalizzazione degli URL Amazon S3 con record CNAME

A seconda delle esigenze, è possibile che non desideri che `s3.region-code.amazonaws.com` venga visualizzato sul tuo sito Web o sul tuo servizio. Se ad esempio ospiti le immagini del sito Web su Amazon S3, è possibile che tu preferisca `http://images.example.com/` anziché `http://images.example.com.s3.us-east-1.amazonaws.com/`. È possibile fare riferimento a qualsiasi bucket con un nome compatibile con DNS come indicato di seguito: `http://BucketName.s3.Region.amazonaws.com/[Filename]`, ad esempio, `http://images.example.com.s3.us-east-1.amazonaws.com/mydog.jpg`. Utilizzando CNAME, è possibile associare `images.example.com` a un nome host Amazon S3 in modo che l'URL precedente diventi `http://images.example.com/mydog.jpg`.

Il nome del bucket deve corrispondere esattamente al CNAME. Ad esempio, se crei un CNAME per associare `images.example.com` a `images.example.com.s3.us-east-1.amazonaws.com`, `http://images.example.com/filename` e `http://images.example.com.s3.us-east-1.amazonaws.com/filename` saranno identici.

Il record DNS di CNAME dovrebbe assegnare al nome del dominio il nome host in stile hosting virtuale appropriato come alias. Se ad esempio il nome del bucket e il nome del dominio sono `images.example.com` e il bucket si trova nella regione Stati Uniti orientali (Virginia settentrionale), il record CNAME dovrebbe assegnare l'alias a `images.example.com.s3.us-east-1.amazonaws.com`.

```
images.example.com CNAME    images.example.com.s3.us-east-1.amazonaws.com.
```

Amazon S3 utilizza il nome host per determinare il nome del bucket. Quindi il nome del bucket e CNAME devono essere gli stessi. Si supponga ad esempio di avere configurato `www.example.com` come CNAME per `www.example.com.s3.us-east-1.amazonaws.com`. Quando accedi a `http://www.example.com`, Amazon S3 riceve una richiesta simile alla seguente:

Example

```
GET / HTTP/1.1
Host: www.example.com
Date: date
Authorization: signatureValue
```

Amazon S3 vede solo il nome host originale `www.example.com` e non rileva la mappatura CNAME usata per risolvere la richiesta.

È possibile utilizzare qualsiasi endpoint Amazon S3 in un alias CNAME. Ad esempio, `s3.ap-southeast-1.amazonaws.com` può essere utilizzato negli alias CNAME. Per ulteriori informazioni sugli endpoint, consulta Endpoint della richiesta. Per creare un sito Web statico utilizzando un dominio personalizzato, consulta Tutorial: Configurazione di un sito Web statico utilizzando un dominio personalizzato registrato con Route 53

> **Important**
>
> Quando si utilizzano URL personalizzati con CNAME, è necessario assicurarsi che esista un bucket corrispondente per qualsiasi record o alias CNAME configurato. Ad esempio, se si creano voci DNS per `www.example.com` e `login.example.com` per pubblicare contenuti Web utilizzando S3, è necessario creare entrambi i bucket `www.example.com` e `login.example.com`.
>
> Quando un record o un alias CNAME è configurato in modo che punti a un endpoint S3 senza un bucket corrispondente, qualsiasi utente AWS può creare quel bucket e pubblicare contenuti con l'alias configurato, anche se la proprietà non è la stessa.
>
> Per lo stesso motivo, si consiglia di modificare o rimuovere il CNAME o l'alias corrispondente quando si elimina un bucket.

Come associare un nome host a un bucket Amazon S3

Associazione di un nome host a un bucket Amazon S3 utilizzando un alias CNAME

1. Selezionare un nome host appartenente a un dominio sottoposto al proprio controllo.

   In questo esempio viene utilizzato il sottodominio `images` del dominio `example.com`.

2. Creare un bucket che corrisponda al nome host.

   In questo esempio i nomi host e del bucket sono `images.example.com`. Il nome del bucket deve corrispondere esattamente al nome host.

3. Creare un record DNS CNAME che definisca il nome host come un alias per il bucket Amazon S3.

   Ad esempio:

```
images.example.com CNAME images.example.com.s3.us-west-2.amazonaws.com
```

> **Important**
>
> Per motivi di instradamento della richiesta, il record DNS del CNAME deve essere definito esattamente come mostrato nell'esempio precedente. In caso contrario, potrebbe mostrare un funzionamento corretto ma determinare un comportamento imprevisto.

La procedura per la configurazione dei record DNS CNAME dipende dal server o dal provider DNS. Per informazioni specifiche, consultare la documentazione del server o contattare il provider.

Restrizioni

Il supporto di SOAP su HTTP non viene più utilizzato ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non sono supportate per SOAP. Invece di utilizzare SOAP, si consiglia di utilizzare REST API o gli SDK AWS.

Compatibilità con le versioni precedenti

Le sezioni seguenti trattano vari aspetti della compatibilità con le versioni precedenti di Amazon S3 relativi a richieste URL in stile percorso e in stile hosting virtuale.

Endpoint legacy

Alcune regioni supportano gli endpoint legacy. Questi endpoint potrebbero essere visualizzati nei log di accesso al server o nei log AWS CloudTrail. Per ulteriori informazioni, consulta le informazioni riportate di seguito. Per un elenco completo degli endpoint e delle regioni Amazon S3, consultare la sezione relativa a endpoint e quote di Amazon S3 nella Riferimenti generali di Amazon Web Services.

> **Important**
>
> Sebbene nei log siano presenti endpoint legacy, si consiglia di utilizzare sempre la sintassi standard dell'endpoint per accedere ai bucket.
> Gli URL in stile hosting virtuale di Amazon S3 utilizzano il seguente formato.
>
> ```
> https://bucket-name.s3.region-code.amazonaws.com/key-name
> ```

In Amazon S3, gli URL in stile percorso utilizzano il seguente formato.

```
https://s3.region-code.amazonaws.com/bucket-name/key-name
```

s3-Regione

Alcune regioni Amazon S3 meno recenti supportano endpoint che contengono un trattino (`-`) tra `s3` e il codice della regione (ad esempio, `s3-us-west-2`) anziché un punto (ad esempio, `s3.us-west-2`). Se il bucket si trova in una di queste regioni, potrebbe essere visualizzato il seguente formato endpoint nei log di accesso al server o nei log di CloudTrail:

```
https://bucket-name.s3-region-code.amazonaws.com
```

In questo esempio, il nome del bucket è DOC-EXAMPLE-BUCKET1 e la regione è Stati Uniti occidentali (Oregon):

```
https://DOC-EXAMPLE-BUCKET1.s3-us-west-2.amazonaws.com
```

Endpoint globale legacy

Per alcune regioni, è possibile utilizzare l'endpoint globale legacy per costruire richieste che non specificano un endpoint specifico della regione. L'endpoint globale legacy è il seguente:

```
bucket-name.s3.amazonaws.com
```

Nei log di accesso al server o nei log di CloudTrail, è possibile che vengano visualizzate richieste che utilizzano l'endpoint globale legacy. In questo esempio, il nome del bucket è `DOC-EXAMPLE-BUCKET1` e l'endpoint globale legacy è:

```
https://DOC-EXAMPLE-BUCKET1.s3.amazonaws.com
```

Richieste virtuali in stile hosting virtuale per Stati Uniti orientali (Virginia settentrionale)

Le richieste effettuate con l'endpoint globale legacy sono instradate negli Stati Uniti orientali (Virginia settentrionale) per impostazione predefinita. Pertanto, l'endpoint globale legacy viene talvolta utilizzato al posto dell'endpoint regionale per Stati Uniti orientali (Virginia settentrionale). Se crei un

bucket in Stati Uniti orientali (Virginia settentrionale) e utilizzi l'endpoint globale, Amazon S3 instrada la richiesta a questa regione per impostazione predefinita.

Richieste in stile hosting virtuale per altre regioni

L'endpoint globale legacy viene utilizzato anche per le richieste in stile hosting virtuale nelle altre regioni supportate. Se crei un bucket in una regione lanciata prima del 20 marzo 2019 e utilizzi l'endpoint globale legacy, Amazon S3 aggiorna il record DNS per reinstradare la richiesta alla posizione corretta. Questa operazione potrebbe richiedere del tempo. Nel frattempo, viene applicata la regola di default: la richiesta in stile hosting virtuale viene inviata alla regione Stati Uniti orientali (Virginia settentrionale). Amazon S3 quindi la reindirizza con un reindirizzamento HTTP 307 temporaneo alla regione corretta.

Per i bucket S3 in regioni lanciate dopo il 20 marzo 2019, il server DNS non indirizza la richiesta direttamente alla Regione AWS in cui risiede il bucket. Restituisce invece un errore HTTP 400 - Richiesta non valida. Per ulteriori informazioni, consulta Esecuzione di richieste.

Richieste in stile percorso

Per la regione Stati Uniti orientali (Virginia settentrionale), è possibile utilizzare l'endpoint globale legacy per le richieste in stile percorso.

Per tutte le altre regioni, la sintassi in stile percorso richiede l'utilizzo dell'endpoint specifico della regione quando si cerca di accedere al bucket. Se si tenta di accedere a un bucket con l'endpoint globale legacy o un altro endpoint diverso rispetto a quello della regione in cui risiede il bucket, viene visualizzato un errore 307 Temporary Redirect del codice di risposta HTTP che indica qual è l'URI corretto per la risorsa. Ad esempio, se utilizzi `https://s3.amazonaws.com/`*`bucket-name`* per un bucket creato nella regione Stati Uniti occidentali (Oregon), verrà visualizzato un errore di reindirizzamento temporaneo HTTP 307.

Reindirizzamento delle richieste e API REST

Argomenti

- Reindirizzamenti e utenti-agenti HTTP
- Reindirizzamenti e 100-continue
- Esempio di reindirizzamento

Questa sezione descrive come gestire i reindirizzamenti HTTP tramite l'API REST di Amazon S3. Per informazioni generali sui reindirizzamenti di Amazon S3, consulta Esecuzione di richieste nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Reindirizzamenti e utenti-agenti HTTP

I programmi che utilizzano l'API REST di Amazon S3 devono gestire i reindirizzamenti a livello di applicazione o a livello di HTTP. Molte librerie di client e utenti-agenti HTTP possono essere configurate per gestire automaticamente i reindirizzamenti in modo corretto; tuttavia, in molti altri casi, le implementazioni sono incorrette o incomplete.

Prima di affidarsi a una libreria per soddisfare il requisito di reindirizzamento, testare i casi riportati di seguito:

- Verificare che la richiesta reindirizzata (la seconda richiesta dopo la ricezione del reindirizzamento) includa tutte le intestazioni della richiesta HTTP, comprese quelle standard relative, ad esempio, ad autorizzazione e data.
- Verificare il corretto funzionamento dei reindirizzamenti non GET, come PUT e DELETE.
- Verificare che le richieste PUT di grandi dimensioni seguano correttamente i reindirizzamenti.
- Se la risposta 100-continue impiega molto tempo ad arrivare, verificare che le richieste PUT seguano correttamente i reindirizzamenti.

Se il metodo di richiesta HTTP non è GET o HEAD, gli utenti-agenti HTTP rigorosamente conformi allo standard RFC 2616 potrebbero richiedere una conferma esplicita prima di seguire un reindirizzamento. In genere è sicuro seguire i reindirizzamenti generati automaticamente da Amazon S3, perché il sistema li emette solo verso host interni al dominio amazonaws.com e l'effetto della richiesta reindirizzata è uguale a quello della richiesta originale.

Reindirizzamenti e 100-continue

Configurare l'applicazione per l'utilizzo di 100-continue per le operazioni PUT consente di semplificare la gestione dei reindirizzamenti, migliorare l'efficienza ed evitare i costi che comporta inviare due volte un corpo della richiesta reindirizzato. Con l'utilizzo di 100-continue, l'applicazione non invia il corpo della richiesta finché non riceve una conferma. Se, in base alle intestazioni, il messaggio viene rifiutato, il corpo del messaggio non viene inviato. Per ulteriori informazioni su 100-continue, consulta RFC 2616 sezione 8.2.3

> **Note**
>
> Secondo lo standard RFC 2616, quando si utilizza `Expect: Continue` con un server HTTP sconosciuto, non bisogna attendere un periodo di tempo indefinito prima inviare il corpo della richiesta, perché alcuni server HTTP non riconoscono 100-continue. Tuttavia, Amazon S3 è in grado di riconoscere se la richiesta contiene un codice `Expect: Continue` e risponderà con uno stato 100-continue provvisorio o con un codice di stato definitivo. Inoltre, dopo aver ricevuto il via libera provvisorio per 100-continue, non si verificheranno errori di reindirizzamento: in questo modo non si riceverà una risposta di reindirizzamento mentre è ancora in corso la scrittura del corpo della richiesta.

Esempio di reindirizzamento

In questa sezione è riportato un esempio di interazione fra client e server con utilizzo di reindirizzamenti HTTP e 100-continue.

Di seguito è riportato un esempio di operazione PUT nel 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 restituisce quanto segue:

```
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>
```

Il client segue la risposta di reindirizzamento ed emette una nuova richiesta per l'endpoint temporaneo `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 restituisce un 100-continue indicando al client di procedere con l'invio del corpo della richiesta.

```
HTTP/1.1 100 Continue
```

Il client invia il corpo della richiesta.

```
ha ha\n
```

Amazon S3 restituisce la risposta definitiva.

```
HTTP/1.1 200 OK
Date: Mon, 15 Oct 2007 22:18:48 GMT

ETag: "a2c8d6b872054293afd41061e93bc289"
Content-Length: 0
Server: AmazonS3
```

Sviluppo con Amazon S3 tramite la AWS CLI

Segui la procedura per scaricare e configurare AWS Command Line Interface (AWS CLI).

Per un elenco dei comandi AWS CLI di Amazon S3, consulta le seguenti pagine in Riferimento ai comandi della AWS CLI:

- s3
- s3api
- s3control

> **Note**
>
> Per accedere ai servizi in AWS, come Amazon S3, è necessario fornire le credenziali. Il servizio può quindi stabilire se si dispone delle autorizzazioni per accedere alle risorse del servizio stesso. Per la console è necessaria la password. Per consentire all'Account AWS di accedere a AWS CLI o all'API, è possibile creare chiavi di accesso. È sconsigliabile tuttavia accedere ad AWS utilizzando le credenziali dell'Account AWS. Consigliamo di utilizzare invece AWS Identity and Access Management (IAM). Crea un utente IAM, aggiungilo a un gruppo IAM con autorizzazioni amministrative, quindi concedi le autorizzazioni amministrative all'utente IAM creato. Puoi quindi accedere ad AWS usando un URL speciale e le credenziali di tale utente IAM. Per istruzioni, consulta Creazione del primo utente IAM e del gruppo di amministratori nella Guida per l'utente di IAM.

Per configurare AWS CLI

1. Scarica e configura la AWS CLI. Per istruzioni, consulta i seguenti argomenti nella Guida per l'utente dell'AWS Command Line Interface:
   - Come configurare la AWS Command Line Interface
   - Configurazione della AWS Command Line Interface
2. Aggiungi un profilo denominato per l'utente amministratore nel file di configurazione di AWS CLI. Puoi usare questo profilo quando esegui i comandi AWS CLI. Per ulteriori informazioni, consulta Named profiles for the AWS CLI nella Guida per l'utente di AWS Command Line Interface.

```
[adminuser]
aws_access_key_id = adminuser access key ID
aws_secret_access_key = adminuser secret access key
region = aws-region
```

Per un elenco di Regioni AWS disponibili, consultare la sezione relativa a regioni ed endpoint nella Riferimenti generali di AWS.

3. Verificare la configurazione digitando i seguenti comandi al prompt dei comandi.

   - Provare il comando `help` per verificare che il servizio AWS CLI sia installato sul computer:

     ```
     aws help
     ```

   - Eseguire un comando `S3` usando le credenziali `adminuser` appena create. A tale scopo, aggiungere il parametro `--profile` al comando per specificare il nome del profilo. In questo esempio, il comando `ls` elenca tutti i bucket nell'account. AWS CLI utilizza le credenziali dell'utente `adminuser` per autenticare la richiesta.

     ```
     aws s3 ls --profile adminuser
     ```

Sviluppo con Amazon S3 utilizzando gli SDK AWS

AWS I kit di sviluppo software (SDK) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice, e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

> **Note**
>
> È possibile utilizzarlo AWS Amplify per lo sviluppo end-to-end completo di app Web e mobili. Amplify Storage integra perfettamente le funzionalità di archiviazione e gestione dei file in app web e mobili frontend, basate su Amazon S3. Per ulteriori informazioni, consulta Storage nella guida per l'utente di Amplify.

Utilizzo di questo servizio con un SDK AWS

AWS I kit di sviluppo software (SDK) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice, e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

Documentazione sugli SDK	Esempi di codice
AWS SDK for C++	AWS SDK for C++ esempi di codice
AWS CLI	AWS CLI esempi di codice
AWS SDK for Go	AWS SDK for Go esempi di codice
AWS SDK for Java	AWS SDK for Java esempi di codice
AWS SDK for JavaScript	AWS SDK for JavaScript esempi di codice
SDK AWS for Kotlin	SDK AWS for Kotlin esempi di codice
AWS SDK for .NET	AWS SDK for .NET esempi di codice
AWS SDK for PHP	AWS SDK for PHP esempi di codice
AWS Tools for PowerShell	Strumenti per esempi di PowerShell codice
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) esempi di codice
AWS SDK for Ruby	AWS SDK for Ruby esempi di codice
AWS SDK for Rust	AWS SDK for Rust esempi di codice
SDK AWS per SAP ABAP	SDK AWS per SAP ABAP esempi di codice
SDK AWS per Swift	SDK AWS per Swift esempi di codice

Per esempi specifici del servizio, consulta Esempi di codice per Amazon S3 che utilizzano SDK AWS.

> **Esempio di disponibilità**
>
> Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link Provide feedback (Fornisci un feedback) nella parte inferiore di questa pagina.

interfacce di programmazione SDK

Ogni AWS SDK fornisce una o più interfacce programmatiche per lavorare con Amazon S3. Ogni SDK fornisce un'interfaccia di basso livello per Amazon S3, con metodi molto simili alle operazioni API. Alcuni SDK forniscono interfacce di alto livello per Amazon S3, che sono astrazioni destinate a semplificare i casi d'uso comuni.

Ad esempio, quando esegui un caricamento in più parti utilizzando le operazioni API di basso livello, devi utilizzare un'operazione per avviare il caricamento, un'altra operazione per caricare parti e un'operazione finale per completare il caricamento. Un'operazione API di caricamento multiparte di alto livello consente di eseguire tutte le operazioni necessarie per il caricamento in un'unica chiamata API. Per alcuni esempi, consulta Caricamento di un oggetto utilizzando il caricamento in più parti.

Le operazioni API di basso livello consentono un maggiore controllo sul caricamento. Ti consigliamo di utilizzare le operazioni API di basso livello se devi mettere in pausa e riprendere i caricamenti, variare le dimensioni delle parti durante il caricamento o iniziare i caricamenti quando non conosci in anticipo le dimensioni dei dati.

Specifica di Signature Version nell'autenticazione delle richieste

Nella maggior parte dei casi, Amazon S3 supporta solo AWS la versione 4 di Signature. Regioni AWS In alcuni dei modelli precedenti Regioni AWS, Amazon S3 supporta sia Signature Version 4 che Signature Version 2. Comunque, Signature Version 2 sta per essere disattivato perché obsoleto. Per ulteriori informazioni sulla fine del supporto per Signature Version 2, consulta AWS Signature versione 2 disattivata (obsoleta) per Amazon S3.

Per un elenco di tutte le regioni Amazon S3 e delle versioni di firma supportate, consulta Regioni ed endpoint in Riferimenti generali AWS .

Per tutti Regioni AWS, gli AWS SDK utilizzano per impostazione predefinita la versione 4 di Signature per autenticare le richieste. Quando utilizzi AWS SDK rilasciati prima di maggio 2016, potrebbe esserti richiesto di richiedere la versione 4 di Signature, come mostrato nella tabella seguente.

SDK	Richiesta di Signature Version 4 per autenticazione delle richieste
AWS CLI	Per il profilo di default, eseguire il comando indicato di seguito:

SDK	Richiesta di Signature Version 4 per autenticazione delle richieste
	`$ aws configure set default.s3.signature_version s3v4` Per un profilo personalizzato, eseguire il comando indicato di seguito: `$ aws configure set profile.your_profile_name.s3.signature_version s3v4`
SDK Java	Aggiungere quanto segue al codice: `System.setProperty(SDKGlobalConfiguration.ENABLE_S3_SIGV4_SYSTEM_PROPERTY, "true");` In alternativa, specificare quanto segue nella riga di comando: `-Dcom.amazonaws.services.s3.enableV4`
JavaScript SDK	Impostare il parametro `signatureVersion` su `v4` durante la creazione del client: `var s3 = new AWS.S3({signatureVersion: 'v4'});`

SDK	Richiesta di Signature Version 4 per autenticazione delle richieste
SDK PHP	Impostare il parametro `signature` su `v4` durante la creazione del client del servizio Amazon S3 per l'SDK PHP v2: `<?php` `$client = S3Client::factory([` `'region' => 'YOUR-REGION',` `'version' => 'latest',` `'signature' => 'v4'` `]);` Quando si usa l'SDK PHP v3, impostare il parametro `signature_version` su `v4` durante la creazione del client del servizio Amazon S3: `<?php` `$s3 = new Aws\S3\S3Client([` `'version' => '2006-03-01',` `'region' => 'YOUR-REGION',` `'signature_version' => 'v4'` `]);`
SDK Python-Boto	Specificare quanto segue nel file di configurazione boto di default: `[s3] use-sigv4 = True`

SDK	Richiesta di Signature Version 4 per autenticazione delle richieste
SDK Ruby	SDK Ruby - Versione 1: impostare il parametro `:s3_signature_version` su `:v4` durante la creazione del client: `s3 = AWS::S3::Client.new(:s3_signature_version => :v4)` SDK Ruby - Versione 3: impostare il parametro `signature_version` su `v4` durante la creazione del client: `s3 = Aws::S3::Client.new(signature_version: 'v4')`
SDK .NET	Aggiungere quanto segue al codice prima di creare il client Amazon S3: `AWSConfigsS3.UseSignatureVersion4 = true;` In alternativa, aggiungere quanto segue al file di configurazione: `<appSettings> <add key="AWS.S3.UseSignatureVersion4" value="true" /> </appSettings>`

AWS Signature versione 2 disattivata (obsoleta) per Amazon S3

Signature Version 2 sta per essere disattivato perché obsoleto in Amazon S3. In seguito, Amazon S3 accetterà solo richieste API firmate con Signature Version 4.

Questa sezione fornisce le risposte alle domande più frequenti sulla fine del supporto di Signature Version 2.

Che cos'è Signature Version 2/4 e cosa significa "firmare una richiesta"?

Il processo di firma Signature Version 2 o Signature Version 4 viene utilizzato per autenticare le richieste API Amazon S3. La firma delle richieste consente ad Amazon S3 di identificare il mittente della richiesta e protegge le richieste dalle azioni di utenti malintenzionati.

Per ulteriori informazioni sulla firma delle AWS richieste, consulta Signing AWS API Requests nel. Riferimenti generali di AWS

Quali aggiornamenti vengono introdotti?

Attualmente, sono supportate le richieste API Amazon S3 firmate con i processi Signature Version 2 e Signature Version 4. In seguito, Amazon S3 accetterà solo le richieste firmate con Signature Version 4.

Per ulteriori informazioni sulla firma AWS delle richieste, vedere Modifiche nella versione 4 di Signature nel Riferimenti generali di AWS.

Perché è stato rilasciato questo aggiornamento?

Signature Version 4 fornisce una maggiore sicurezza perché utilizza una chiave di firma invece della chiave di accesso segreta. La versione 4 di Signature è attualmente supportata in tutti Regioni AWS, mentre la versione Signature 2 è supportata solo nelle regioni lanciate prima di gennaio 2014. Questo aggiornamento consente di offrire un'esperienza più uniforme in tutte le regioni.

Come posso verificare di stare utilizzando Signature Version 4 e di quali aggiornamenti ho bisogno?

La versione della firma utilizzata per firmare le richieste viene generalmente impostata dallo strumento o dall'SDK sul lato client. Per impostazione predefinita, le versioni più recenti dei nostri AWS SDK utilizzano Signature Version 4. Per i software di terza parte, contatta il team di supporto specifico per il tuo software per verificare la versione richiesta. Se stai inviando chiamate REST ad Amazon S3, devi modificare l'applicazione per utilizzare il processo di firma Signature Version 4.

Per informazioni sulla versione degli AWS SDK da utilizzare quando si passa alla versione 4 di Signature, consulta. Passaggio da Signature Version 2 a Signature Version 4

Per ulteriori informazioni sull'uso di Signature Version 4 con l'API REST di Amazon S3, consulta Autenticazione delle richieste (AWS Signature Version 4) nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Cosa succede se non eseguo gli aggiornamenti?

Le richieste firmate con Signature Version 2 effettuate successivamente non verranno autenticate con Amazon S3. I richiedenti visualizzeranno degli errori che indicano che la richiesta deve essere firmata con Signature Version 4.

Devo apportare queste modifiche anche se sto utilizzando un URL prefirmato che richiede la firma per un periodo superiore a 7 giorni?

Se utilizzi un URL prefirmato che richiede la firma per un periodo superiore a 7 giorni, non è necessaria alcuna operazione. Puoi continuare a utilizzare AWS Signature Version 2 per firmare e autenticare l'URL predefinito. Più avanti forniremo ulteriori informazioni su come eseguire la migrazione a Signature Version 4 in uno scenario di URL prefirmato.

Ulteriori informazioni

- Per ulteriori informazioni sull'utilizzo della versione 4 di Signature, consulta Signing AWS API Requests.
- Visualizza l'elenco delle differenze tra Signature Version 2 e Signature Version 4 in Modifiche di Signature Version 4.
- Visualizza il post AWS Signature Version 4 per sostituire AWS Signature Version 2 per la firma delle richieste API Amazon S3 nei AWS forum.
- Per domande o dubbi, contatta AWS Support.

Passaggio da Signature Version 2 a Signature Version 4

Se al momento utilizzi Signature Version 2 per l'autenticazione delle richieste API Amazon S3, devi passare a Signature Version 4. Il supporto per Signature Version 2 sta per terminare, come descritto in AWS Signature versione 2 disattivata (obsoleta) per Amazon S3.

Per ulteriori informazioni sull'uso di Signature Version 4 con l'API REST di Amazon S3, consulta Autenticazione delle richieste (AWS Signature Version 4) nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

La tabella seguente elenca gli SDK con la versione minima necessaria per utilizzare Signature Version 4 (SigV4). Se utilizzi URL predefiniti con gli SDK AWS Java, JavaScript (Node.js) o Python (BOTO/CLI), devi impostare la versione 4 di Signature corretta Regione AWS e impostarla nella configurazione del client. Per informazioni sull'impostazione di `SigV4` nella configurazione del client, consulta Specifica di Signature Version nell'autenticazione delle richieste.

Se utilizzi questo SDK/ prodotto	Esegui l'aggiornamento a questa versione dell'SDK	La modifica del codice nel client è necessaria per utilizzare Sigv4?	Collegamento alla documentazione sugli SDK
AWS SDK for Java v1	Aggiornamento a Java 1.11.201+ o v2.	Sì	Specifica di Signature Version nell'autenticazione delle richieste
AWS SDK for Java v2	Non è necessario l'aggiornamento dell'SDK.	No	AWS SDK for Java
AWS SDK for .NET v1	Aggiornamento alla versione 3.1.10 o successiva.	Sì	AWS SDK for .NET
AWS SDK for .NET v2	Aggiornamento alla versione 3.1.10 o successiva.	No	AWS SDK for .NET v2
AWS SDK for .NET v3	Aggiornamento alla versione 3.3.0.0 o successiva.	Sì	AWS SDK for .NET v3
AWS SDK for JavaScript v1	Aggiornamento alla	Sì	AWS SDK for JavaScript

Se utilizzi questo SDK/ prodotto	Esegui l'aggiornamento a questa versione dell'SDK	La modifica del codice nel client è necessaria per utilizzare Sigv4?	Collegamento alla documentazione sugli SDK
	versione 2.68.0 o successiva.		
AWS SDK for JavaScript v2	Aggiornamento alla versione 2.68.0 o successiva.	Sì	AWS SDK for JavaScript
AWS SDK for JavaScript v3	Al momento, non è richiesta alcuna operazione. Aggiornamento alla versione principale V3 nel T3 2019.	No	AWS SDK for JavaScript

Se utilizzi questo SDK/ prodotto	Esegui l'aggiornamento a questa versione dell'SDK	La modifica del codice nel client è necessaria per utilizzare Sigv4?	Collegamento alla documentazione sugli SDK
AWS SDK for PHP v1	Consigliato l'aggiornamento alla versione più recente di PHP o almeno alla v2.7.4 con il parametro di firma impostato su v4 nella configurazione del client S3.	Sì	AWS SDK for PHP

Se utilizzi questo SDK/ prodotto	Esegui l'aggiornamento a questa versione dell'SDK	La modifica del codice nel client è necessaria per utilizzare Sigv4?	Collegamento alla documentazione sugli SDK
AWS SDK for PHP v2	Consigliato l'aggiornamento alla versione più recente di PHP o almeno alla v2.7.4 con il parametro di firma impostato su v4 nella configurazione del client S3.	No	AWS SDK for PHP
AWS SDK for PHP v3	Non è necessario l'aggiornamento dell'SDK.	No	AWS SDK for PHP
Boto2	Aggiornamento a Boto2 v2.49.0.	Sì	Aggiornamento a Boto 2

Se utilizzi questo SDK/ prodotto	Esegui l'aggiornamento a questa versione dell'SDK	La modifica del codice nel client è necessaria per utilizzare Sigv4?	Collegamento alla documentazione sugli SDK
Boto3	Aggiornamento alla versione 1.5.71 (Botocore), 1.4.6 (Boto3).	Sì	Boto 3 - AWS SDK per Python
AWS CLI	Aggiornamento alla versione 1.11.108.	Sì	AWS Command Line Interface
AWS CLI v2 (anteprima)	Non è necessario l'aggiornamento dell'SDK.	No	AWS Command Line Interface versione 2
AWS SDK for Ruby v1	Aggiornamento a Ruby V3.	Sì	Ruby V3 per AWS
AWS SDK for Ruby v2	Aggiornamento a Ruby V3.	Sì	Ruby V3 per AWS
AWS SDK for Ruby v3	Non è necessario l'aggiornamento dell'SDK.	No	Ruby V3 per AWS

Se utilizzi questo SDK/ prodotto	Esegui l'aggiorn amento a questa versione dell'SDK	La modifica del codice nel client è necessaria per utilizzare Sigv4?	Collegamento alla documentazione sugli SDK
Go	Non è necessari o l'aggiorn amento dell'SDK.	No	AWS SDK for Go
C++	Non è necessari o l'aggiorn amento dell'SDK.	No	AWS SDK for C++

AWS Tools for Windows PowerShell oppure AWS Tools for PowerShell Core

Se utilizzi versioni del modulo precedenti alla 3.3.0.0, devi eseguire l'aggiornamento alla versione 3.3.0.0.

Per ottenere informazioni sulla versione, utilizza il cmdlet `Get-Module`:

```
Get-Module -Name AWSPowershell
Get-Module -Name AWSPowershell.NetCore
```

Per eseguire l'aggiornamento alla versione 3.3.0.0, utilizza il cmdlet `Update-Module`:

```
Update-Module -Name AWSPowershell
Update-Module -Name AWSPowershell.NetCore
```

Puoi utilizzare gli URL prefirmati, validi per più di 7 giorni, su cui invierai il traffico di Signature Version 2.

Sviluppo con Amazon S3 utilizzando l'API REST

L'architettura di Amazon S3 è ideata per essere indipendente dal linguaggio di programmazione, utilizzando le nostre interfacce supportate per archiviare e recuperare oggetti.

Amazon S3 attualmente fornisce un'interfaccia REST. Con REST, i metadati vengono restituiti nelle intestazioni HTTP. Poiché sono supportate solo le richieste HTTP di un massimo di 4 KB (senza corpo incluso), è possibile fornire una quantità di metadata limitata. L'API REST è un'interfaccia HTTP per Amazon S3. Con REST, si utilizzano le richieste HTTP standard per creare, recuperare ed eliminare bucket e oggetti.

Per utilizzare l'API REST, è possibile servirsi di qualunque kit di strumenti in grado di supportare HTTP. Puoi anche utilizzare un browser per recuperare gli oggetti, purché siano leggibili in modo anonimo.

Poiché l'API REST utilizza codici di stato e intestazioni HTTP standard, i kit di strumenti e i browser standard funzionano come previsto. In alcune aree sono state aggiunte funzionalità ad HTTP, ad esempio le intestazioni per il supporto del controllo accessi. Le nuove funzionalità sono state in tali casi aggiunte in modo da essere conformi allo stile di utilizzo di HTTP standard.

Per ulteriori informazioni sull'invio di richieste mediante l'API REST, consulta Esecuzione di richieste con l'utilizzo di API REST. Per alcune considerazioni da tenere presente quando si utilizza l'API REST, consulta gli argomenti riportati di seguito.

Per ulteriori informazioni sull'utilizzo della REST API Amazon S3, consulta la Documentazione di riferimento delle API di Amazon Simple Storage Service.

Argomenti

- Instradamento della richiesta

Instradamento della richiesta

I programmi che inviano richieste ai bucket creati utilizzando l'API CreateBucket che includono una CreateBucketConfiguration devono supportare i reindirizzamenti. Inoltre, è possibile che si verifichino problemi con alcuni client che non rispettano i valori TTL DNS.

Questa sezione descrive l'instradamento e i problemi del server DNS da tenere in considerazione durante la progettazione del servizio o dell'applicazione da utilizzare con Amazon S3.

Reindirizzamento delle richieste e API REST

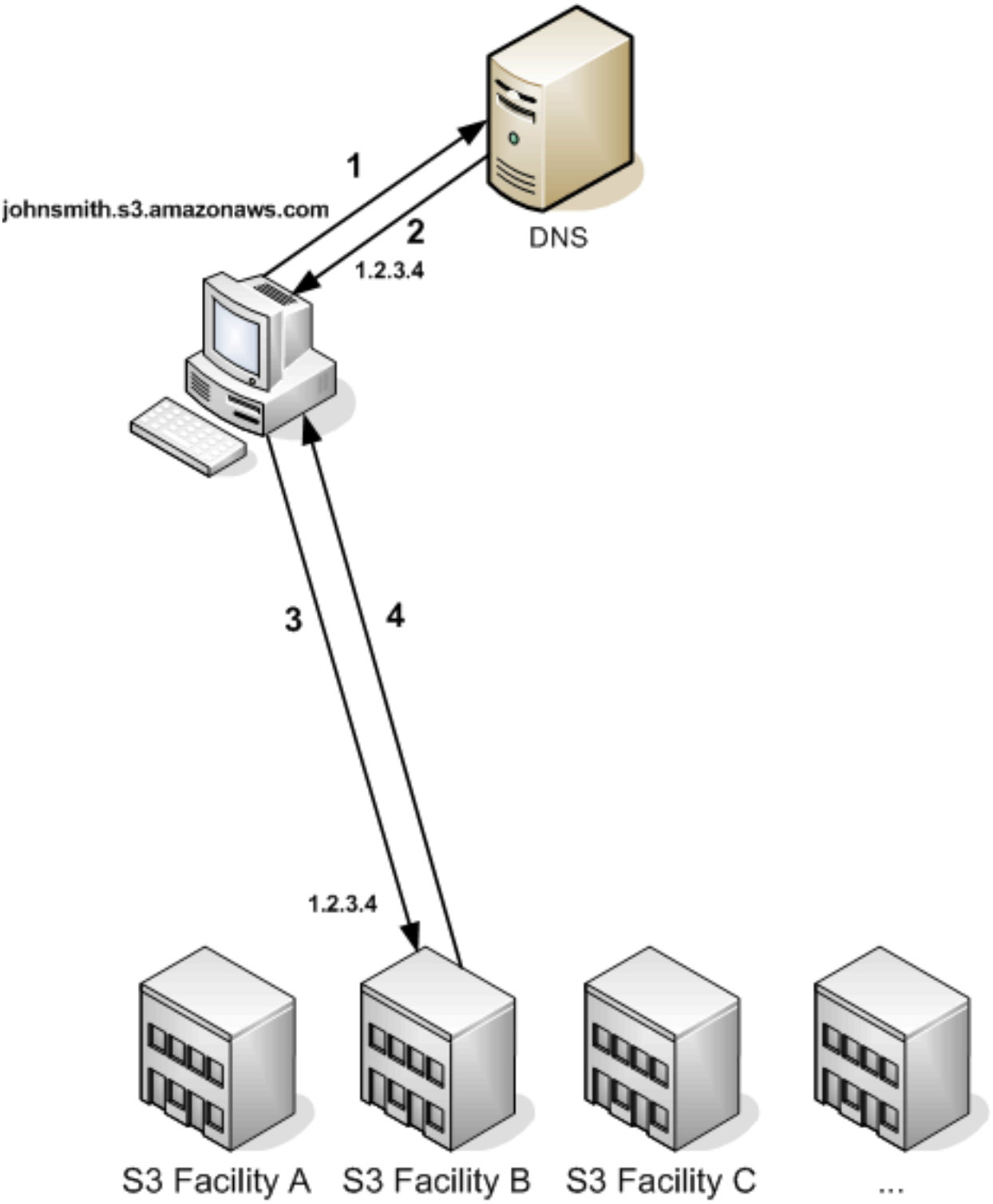
Amazon S3 utilizza Domain Name System (DNS) per instradare le richieste a strutture che possano elaborarle. Il sistema funziona in modo efficace, ma potrebbero verificarsi errori di instradamento temporaneo. Se una richiesta arriva nella posizione Amazon S3 sbagliata, Amazon S3 risponde con un reindirizzamento temporaneo che indica al richiedente di rinviare la richiesta a un altro endpoint. Se la richiesta è formulata in modo errato, Amazon S3 utilizza i reindirizzamenti permanenti per fornire indicazioni su come eseguire la richiesta correttamente.

> **Important**
>
> Per utilizzare questa funzionalità, è necessario essere in possesso di un'applicazione che possa gestire le risposte di reindirizzamento di Amazon S3. L'unica eccezione riguarda i le applicazioni che operano esclusivamente con bucket creati senza `<CreateBucketConfiguration>`. Per ulteriori informazioni sui vincoli di posizione, consulta Accesso ed elenco di un bucket Amazon S3.
>
> Per tutte le regioni lanciate dopo il 20 marzo 2019, se una richiesta arriva alla posizione Amazon S3 errata, Amazon S3 restituisce un errore HTTP 400 - Richiesta non valida.
>
> Per ulteriori informazioni sull'abilitazione o la disabilitazione di una Regione AWS, consultare la sezione relativa a regioni ed endpoint Regioni AWS nella Riferimenti generali di AWS.

Argomenti

Instradamento DNS

L'instradamento DNS indirizza le richieste alle strutture Amazon S3 appropriate. La figura e la procedura seguenti illustrano un esempio di instradamento DNS.



Passaggi della richiesta di instradamento DNS

1. Il client effettua una richiesta DNS per l'archiviazione di un oggetto su Amazon S3.

2. Il client riceve uno o più indirizzi IP per le strutture che possono elaborare la richiesta. In questo esempio, l'indirizzo IP è per la struttura B.
3. Il client effettua una richiesta alla struttura B di Amazon S3.
4. La struttura B restituisce una copia dell'oggetto al client.

Reindirizzamento temporaneo delle richieste

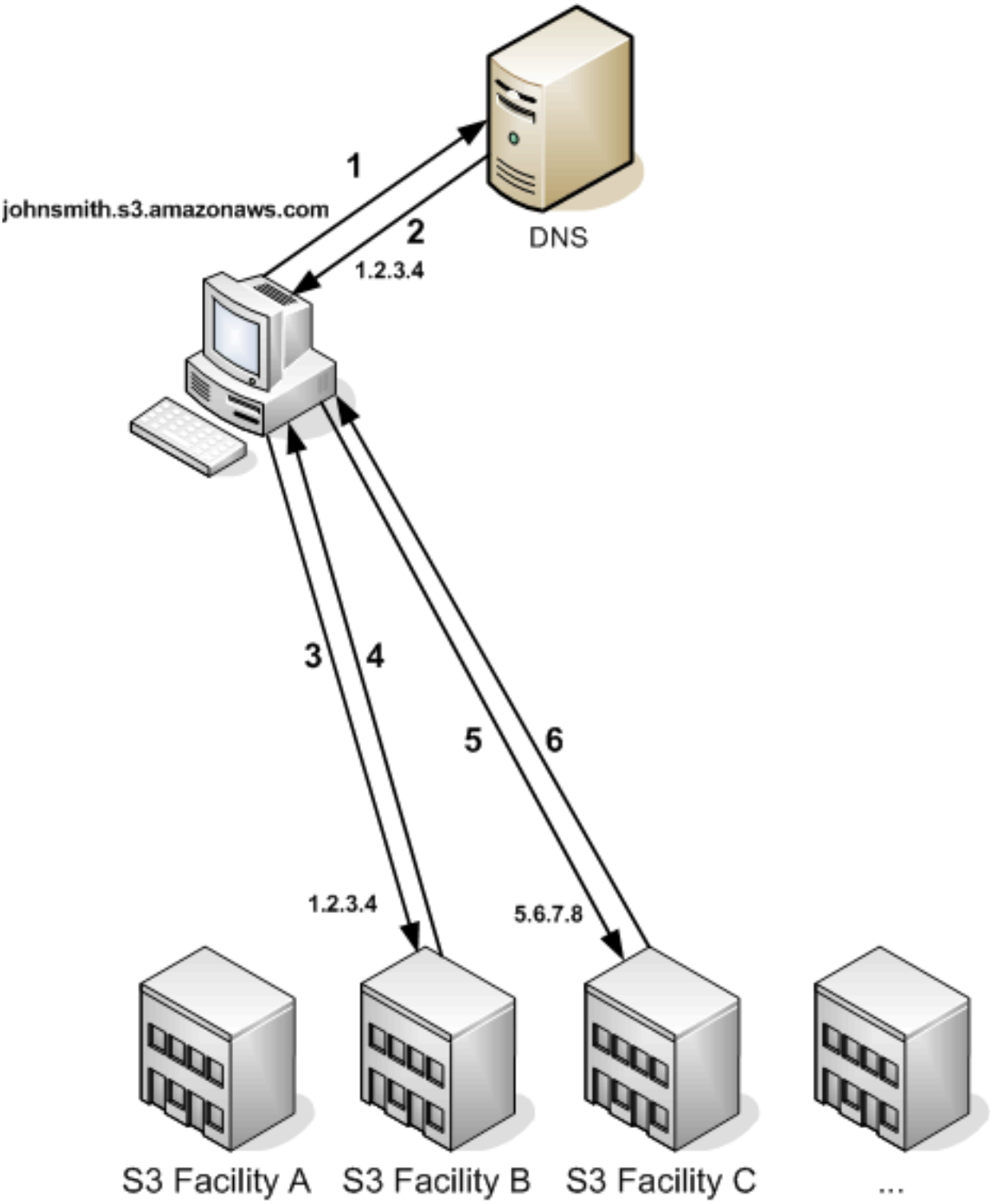
Il reindirizzamento temporaneo è un tipo di risposta di errore che segnala al richiedente la necessità di rinviare la richiesta a un altro endpoint. Data la natura distribuita di Amazon S3, le richieste possono essere temporaneamente instradate alla struttura sbagliata. Questo si verifica con maggiore probabilità immediatamente dopo la creazione o l'eliminazione dei bucket.

Ad esempio, se si crea un nuovo bucket e si effettua immediatamente una richiesta al bucket, si potrebbe ricevere un reindirizzamento temporaneo, a seconda del vincolo di posizione del bucket. Se il bucket è stato creato nella Regione AWS Stati Uniti orientali (Virginia settentrionale), il reindirizzamento non si verifica poiché si tratta anche dell'endpoint Amazon S3 predefinito.

Tuttavia, se il bucket viene creato in qualsiasi altra regione, tutte le richiesta al bucket arrivano all'endpoint predefinito durante la propagazione della voce DNS del bucket. L'endpoint predefinito reindirizza la richiesta all'endpoint corretto con una risposta HTTP 302. I reindirizzamenti temporanei contengono un URI per la struttura corretta, che può essere utilizzato per rinviare immediatamente la richiesta.

> **Important**
>
> Non riutilizzare un endpoint fornito da una risposta di reindirizzamento precedente. Potrebbe sembrare funzionante (anche per lunghi periodi di tempo), ma potrebbe generare risultati imprevedibili e infine dare esito negativo senza preavviso.

La figura e la procedura seguenti illustrano un esempio di reindirizzamento temporaneo.



Passaggi di reindirizzamento temporaneo delle richieste

1. Il client effettua una richiesta DNS per l'archiviazione di un oggetto su Amazon S3.
2. Il client riceve uno o più indirizzi IP per le strutture che possono elaborare la richiesta.

3. Il client effettua una richiesta alla struttura B di Amazon S3.
4. La struttura B restituisce un reindirizzamento che indica che l'oggetto è disponibile dalla posizione C.
5. Il client rinvia la richiesta alla struttura C.
6. La struttura C restituisce una copia dell'oggetto.

Reindirizzamento permanente delle richieste

Un reindirizzamento permanente indica che la richiesta è stata rivolta a una risorsa in modo non appropriato. Ad esempio, i reindirizzamenti permanenti si verificano in caso di utilizzo di una richiesta in stile percorso per accedere a un bucket creato utilizzando `<CreateBucketConfiguration>`. Per ulteriori informazioni, consulta Accesso ed elenco di un bucket Amazon S3.

Per agevolare l'individuazione di questi errori durante lo sviluppo, questo tipo di reindirizzamento non contiene un'intestazione HTTP di posizione che permette di seguire automaticamente la richiesta verso la posizione corretta. Consulta il documento di errore XML risultante come supporto per utilizzare l'endpoint Amazon S3 corretto.

Esempi di reindirizzamento delle richieste

I seguenti sono esempi di risposte di reindirizzamento temporaneo delle richieste.

Risposta di reindirizzamento temporaneo 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>
```

Risposta di reindirizzamento temporaneo API SOAP

> **Note**
>
> Il supporto di SOAP su HTTP non viene più utilizzato ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non sono supportate per SOAP. Invece di utilizzare SOAP, si consiglia di utilizzare REST API o gli SDK AWS.

```
<soapenv:Body>
  <soapenv:Fault>
    <Faultcode>soapenv:Client.TemporaryRedirect</Faultcode>
    <Faultstring>Please re-send this request to the specified temporary endpoint.
    Continue to use the original request endpoint for future requests.</Faultstring>
    <Detail>
      <Bucket>images</Bucket>
      <Endpoint>s3-gztb4pa9sq.amazonaws.com</Endpoint>
    </Detail>
  </soapenv:Fault>
</soapenv:Body>
```

Considerazioni sul sistema DNS

Uno dei requisiti di progettazione di Amazon S3 è una disponibilità estremamente alta. Per rispondere a questo requisito, è possibile aggiornare gli indirizzi IP associati all'endpoint Amazon S3 in DNS, in base alle necessità. Queste modifiche si riflettono automaticamente sui client di breve durata ma non su alcuni client di lunga durata. Per trarre beneficio dalle modifiche apportate, i client di lunga durata dovranno intervenire sull'endpoint Amazon S3 periodicamente. Per ulteriori informazioni sulle macchine virtuali (VM), consultare:

- Per Java, JVM Sun memorizza nella cache le ricerche DNS in modo permanente per impostazione predefinita. Per informazioni su come modificare questo comportamento, consulta la sezione relativa al caching InetAddress della documentazione di InetAddress.
- Per PHP, la VM PHP persistente che esegue le configurazioni di distribuzione più diffuse memorizza nella cache le ricerche DNS fino al riavvio della VM. Consulta la documentazione di PHP relativa a getHostByName.

Gestione degli errori REST e SOAP

Argomenti

Questa sezione descrive gli errori REST e SOAP e come gestirli.

> **Note**
>
> Il supporto di SOAP su HTTP non viene più utilizzato ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non sono supportate per SOAP. Invece di utilizzare SOAP, si consiglia di utilizzare REST API o gli SDK AWS.

La risposta di errore REST

Se una richiesta REST genera un errore, la risposta HTTP contiene:

- Un documento di errore XML come corpo della risposta
- Content-Type: application/xml
- Un codice di stato HTTP 3xx, 4xx o 5xx adeguato

Di seguito è riportato un esempio di risposta di errore 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>
```

Per ulteriori informazioni sugli errori di Amazon S3, consultare ErrorCodeList.

Intestazioni di risposta

Di seguenti sono riportate alcune intestazioni di risposta restituite da tutte le operazioni:

- `x-amz-request-id:` ID univoco assegnato a ogni richiesta dal sistema. Nel caso poco probabile che vi siano problemi di Amazon S3, Amazon può basarsi sull'ID per semplificare l'individuazione e la risoluzione del problema.
- `x-amz-id-2:`Token speciale che ci aiuterà nell'individuazione e risoluzione dei problemi.

Risposta di errore

Quando una richiesta Amazon S3 genera un errore, il cliente riceve una risposta di errore. Il formato esatto della risposta di errore è specifico dell'API: la risposta di errore REST, ad esempio, è diversa da quella di errore SOAP. Tuttavia, tutte le risposte di errore hanno alcuni elementi in comune.

> **Note**
>
> Il supporto di SOAP su HTTP non viene più utilizzato ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non sono supportate per SOAP. Invece di utilizzare SOAP, si consiglia di utilizzare REST API o gli SDK AWS.

Codice di errore

Il codice di errore è una stringa che identifica in modo univoco una condizione di errore. Ha lo scopo di essere letta e compresa dai programmi che rilevano e gestiscono gli errori in base al loro tipo. Molti codici di errore sono condivisi dalle API SOAP e REST, ma alcuni sono specifici dell'API. Ad esempio, l'errore NoSuchKey è universale, ma l'errore UnexpectedContent può verificarsi solo in risposta a una richiesta REST non valida. In tutti i casi, i codici di errore SOAP recano i prefissi indicati nella tabella dei codici di errore, cosicché un errore NoSuchKey viene in realtà restituito in SOAP come Client.NoSuchKey.

> **Note**
>
> Il supporto di SOAP su HTTP non viene più utilizzato ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non sono supportate per SOAP. Invece di utilizzare SOAP, si consiglia di utilizzare REST API o gli SDK AWS.

Messaggio di errore

Il messaggio di errore contiene una descrizione generica della condizione di errore in inglese. È destinato a interlocutori umani. I programmi semplici visualizzano il messaggio direttamente all'utente finale se si verifica una condizione di errore che non conoscono o che non sono interessati a gestire. I programmi sofisticati, con una gestione degli errori più completa e una vera e propria internazionalizzazione, più probabilmente ignoreranno il messaggio di errore.

Ulteriori dettagli

Molte risposte di errore contengono ulteriori dati strutturati destinati a essere letti e compresi da uno sviluppatore che tenti di diagnosticare gli errori di programmazione. Ad esempio, se si invia un'intestazione Content-MD5 con una richiesta REST PUT che non corrisponde al digest calcolato sul server, si riceverà un errore BadDigest. La risposta di errore include anche il digest che abbiamo calcolato come elementi dettagliati, oltre al digest previsto. Durante lo sviluppo è possibile utilizzare queste informazioni per la diagnostica dell'errore. In produzione, un programma con un comportamento corretto probabilmente includerebbe queste informazioni in un registro di errore.

La risposta di errore SOAP

> **Note**
>
> Il supporto di SOAP su HTTP non viene più utilizzato ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non sono supportate per SOAP. Invece di utilizzare SOAP, si consiglia di utilizzare REST API o gli SDK AWS.

In SOAP, il risultato dell'errore viene restituito al client come errore SOAP, con il codice di risposta HTTP 500. Se non si riceve un errore SOAP, la richiesta è stata eseguita correttamente. Il codice di errore SOAP di Amazon S3 comprende un codice di errore basato sullo standard SOAP 1.1 ("Server" o "Client") concatenato con il codice di errore specifico di Amazon S3. Ad esempio: "Server.InternalError" o "Client.NoSuchBucket". L'elemento stringa di errore di SOAP contiene un messaggio di errore generico in inglese leggibile da un interprete umano. Infine, l'elemento del dettaglio di errore SOAP contiene informazioni varie pertinenti all'errore.

Ad esempio, se si tenta di eliminare l'oggetto "Fred", che non esiste, il corpo della risposta SOAP conterrà un errore SOAP "NoSuchKey".

Example

```
<soapenv:Body>
  <soapenv:Fault>
    <Faultcode>soapenv:Client.NoSuchKey</Faultcode>
    <Faultstring>The specified key does not exist.</Faultstring>
    <Detail>
      <Key>Fred</Key>
    </Detail>
  </soapenv:Fault>
</soapenv:Body>
```

Per ulteriori informazioni sugli errori di Amazon S3, consultare ErrorCodeList.

Best Practice per gli errori Amazon S3

Quando si progetta un'applicazione da utilizzare con Amazon S3, è importante gestire correttamente gli errori Amazon S3. Questa sezione descrive i problemi da tenere in considerazione durante la progettazione dell'applicazione.

Errori interni

Gli errori interni si verificano nell'ambiente Amazon S3.

Le richieste che ricevono una risposta di InternalError potrebbero non essere state elaborate. Ad esempio, se una richiesta PUT restituisce InternalError, una GET successiva potrebbe recuperare o il vecchio valore o quello aggiornato.

Se Amazon S3 restituisce una risposta di InternalError, provare a inviare di nuovo la richiesta.

Regolare l'applicazione in caso di errore di rallentamento ripetuto

Come con ogni sistema distribuito, S3 è dotato di meccanismi di protezione che rilevano il consumo eccessivo di risorse, voluto o accidentale, e rispondono di conseguenza. Gli errori di SlowDown possono verificarsi quando un tasso elevato di richieste attiva uno di questi meccanismi. La riduzione del tasso delle richieste porterà alla diminuzione o eliminazione degli errori di questo tipo. In generale, la maggior parte degli utenti non riscontrerà questi errori regolarmente; tuttavia, se desideri maggiori informazioni o stai riscontrando errori di SlowDown numerosi o imprevisti, pubblica una richiesta sul nostro forum degli sviluppatori di Amazon S3 o registrati ad AWS Support all'indirizzo https://aws.amazon.com/premiumsupport/.

Errori isolati

> **Note**
>
> Il supporto di SOAP su HTTP non viene più utilizzato ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non sono supportate per SOAP. Invece di utilizzare SOAP, si consiglia di utilizzare REST API o gli SDK AWS.

Amazon S3 fornisce un insieme di codici di errori usati sia dall'API SOAP che dall'API REST. L'API SOAP restituisce codici di errore Amazon S3 standard. L'API REST è progettata per somigliare a un server HTTP standard e interagisce con i client HTTP esistenti (browser, librerie del client HTTP, proxy, cache e così via). Per assicurarci che i client HTTP gestiscano correttamente gli errori, mappiamo ogni errore di Amazon S3 su un codice di stato HTTP.

I codici di stato HTTP sono meno intuitivi dei codici di errore Amazon S3 e contengono un minor numero di informazioni sull'errore. Ad esempio, gli errori di Amazon S3 `NoSuchKey` e `NoSuchBucket` sono mappati entrambi sul codice di stato `HTTP 404 Not Found`.

Sebbene i codici di stato HTTP contengano un minor numero di informazioni sull'errore, i client che comprendono l'HTTP ma non l'API di Amazon S3 in genere gestiscono l'errore correttamente.

Per questo motivo, quando si gestiscono o si riportano errori di Amazon S3 agli utenti finali, è opportuno utilizzare il codice di errore di Amazon S3 anziché il codice di stato HTTP, perché contiene il maggior numero possibile di informazioni sull'errore. Inoltre, quando si esegue il debug dell'applicazione, è bene consulta l'elemento <Details> della risposta di errore XML, che può essere letto da un interlocutore umano.

Riferimento per gli sviluppatori

Questa appendice include le sezioni seguenti.

Argomenti

Appendice A: utilizzo dell'API SOAP

> **Note**
>
> Il supporto di SOAP su HTTP non viene più utilizzato ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non sono supportate per SOAP. Invece di usare SOAP, ti consigliamo di utilizzare l'API REST o gli AWS SDK.

Questa sezione contiene informazioni specifiche sull'API SOAP di Amazon S3.

> **Note**
>
> Le richieste SOAP, autenticate e anonime, devono essere inviate ad Amazon S3 tramite SSL. Amazon S3 restituisce un errore quando invii una richiesta SOAP tramite HTTP.

Argomenti

Elementi comuni dell'API SOAP

> **Note**
>
> Il supporto di SOAP su HTTP non viene più utilizzato ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non sono supportate per SOAP. Invece di utilizzare SOAP, si consiglia di utilizzare REST API o gli SDK AWS.

È possibile interagire con Amazon S3 utilizzando SOAP 1.1 su HTTP. Amazon S3 WSDL, che descrive l'API di Amazon S3 in modo leggibile dai computer, è disponibile all'indirizzo https://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl. Lo schema Amazon S3 è disponibile all'indirizzo https://doc.s3.amazonaws.com/2006-03-01/AmazonS3.xsd.

La maggior parte degli utenti utilizza il kit di strumenti SOAP, ottimizzato per il linguaggio e l'ambiente di sviluppo in uso, per interagire con Amazon S3. Kit di strumenti diversi espongono l'API di Amazon S3 in modi diversi. Per informazioni su come utilizzarla, consultare la documentazione del kit di strumenti specifico. Questa sezione presenta le operazioni SOAP in Amazon S3 in modo indipendente dal kit di strumenti, descrivendo le richieste e le risposte XML così come appaiono "nella rete".

Elementi comuni

In qualsiasi richiesta SOAP è possibile includere i seguenti elementi correlati all'autorizzazione:

- `AWSAccessKeyId:` L'ID chiave di accesso AWS del richiedente
- `Timestamp:` l'ora corrente del sistema
- `Signature:` la firma per la richiesta

Autenticazione delle richieste SOAP

> **Note**
>
> Il supporto di SOAP su HTTP non viene più utilizzato ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non sono supportate per SOAP. Invece di utilizzare SOAP, si consiglia di utilizzare REST API o gli SDK AWS.

Per attestare l'identità dell'entità che effettua la richiesta, ogni richiesta non anonima deve contenere informazioni di autenticazione. In SOAP le informazioni di autenticazione sono inserite nei seguenti elementi della richiesta SOAP:

- L'ID chiave di accesso AWS

  > **Note**
  >
  > Quando si effettuano richieste SOAP autenticate; le credenziali di sicurezza temporanee non sono supportate. Per ulteriori informazioni sui tipi di credenziali, consultare Esecuzione di richieste.

- `Timestamp:` Deve essere un valore dateTime (visita l'indirizzo http://www.w3.org/TR/xmlschema-2/#dateTime) espresso nell'orario UTC (fuso orario di Greenwich), ad esempio

`2009-01-01T12:00:00.000Z`. L'autorizzazione non riuscirà se la differenza tra il timestamp e l'orologio nei server Amazon S3 è maggiore di 15 minuti.

- `Signature:` Il digest HMAC-SHA1 RFC 2104 (http://www.ietf.org/rfc/rfc2104.txt) della concatenazione "AmazonS3" + OPERAZIONE + Timestamp, utilizzando la chiave di accesso segreta AWS come chiave. Nella seguente richiesta di esempio CreateBucket l'elemento Signature (firma) contiene ad esempio il digest HMAC-SHA1 del valore "AmazonS3CreateBucket2009-01-01T12:00:00.000Z":

Nella seguente richiesta di esempio CreateBucket l'elemento Signature (firma) contiene ad esempio il digest HMAC-SHA1 del valore "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

Le richieste SOAP, autenticate e anonime, devono essere inviate ad Amazon S3 tramite SSL. Amazon S3 restituisce un errore quando invii una richiesta SOAP tramite HTTP.

Important

A causa delle diverse interpretazioni riguardo a come ridurre la precisione del tempo aggiuntivo, gli utenti .NET devono evitare di inviare ad Amazon S3 timestamp troppo specifici. Ciò è possibile strutturando manualmente gli oggetti `DateTime` esclusivamente con una precisione pari al millisecondo.

Impostazione della policy d'accesso con SOAP

> **Note**
>
> Il supporto di SOAP su HTTP non viene più utilizzato ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non sono supportate per SOAP. Invece di usare SOAP, ti consigliamo di utilizzare l'API REST o gli AWS SDK.

Il controllo dell'accesso può essere impostato nel momento in cui un bucket o un oggetto viene scritto includendo l'elemento AccessControlList "" nella richiesta a `CreateBucketPutObjectInline`, o. `PutObject` L' AccessControlList elemento è descritto inIdentity and Access Management per Amazon S3. Se non viene specificata alcuna lista di controllo degli accessi con queste operazioni, la risorsa viene creata con una politica di accesso predefinita che fornisce al richiedente l'accesso FULL_CONTROL (questo è il caso anche se la richiesta è una PutObject richiesta PutObjectInline or per un oggetto già esistente).

Di seguito viene riportata una richiesta che scrive dati in un oggetto, lo rende leggibile da parte di entità principali anonime e concede all'utente specificato i diritti FULL_CONTROL sul bucket (la maggior parte degli sviluppatori si assegna l'accesso FULL_CONTROL al bucket di cui sono proprietari).

Example

Di seguito è riportata una richiesta che scrive dati in un oggetto e lo rende leggibile da parte di entità principali anonime.

`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>
```

È possibile leggere o impostare la policy di controllo accessi per un bucket o un oggetto esistente utilizzando i metodi `GetBucketAccessControlPolicy`, `GetObjectAccessControlPolicy`, `SetBucketAccessControlPolicy` e `SetObjectAccessControlPolicy`. Per ulteriori informazioni, consulta la spiegazione dettagliata di questi metodi.

Appendice b: Richieste di autenticazione (versione 2 della firma)AWS

> **Important**
>
> Questa sezione descrive come autenticare le richieste utilizzando AWS Signature Version 2. Signature Version 2 sta per essere disattivato in quanto obsoleto e Amazon S3 accetterà solo richieste API firmate con Signature Version 4. Per ulteriori informazioni, consulta AWS Signature versione 2 disattivata (obsoleta) per Amazon S3

La versione 4 di Signature è supportata in tutte le Regioni AWS aree ed è l'unica versione supportata per le nuove regioni. Per ulteriori informazioni, consulta Authenticating Requests (AWS Signature Version 4) nel riferimento all'API di Amazon Simple Storage Service. Amazon S3 offre la possibilità di identificare il protocollo Signature Version dell'API utilizzato per firmare una richiesta. È importante stabilire se qualcuno dei flussi di lavoro sta usando Signature Versione 2 e aggiornarlo in modo che utilizzi Signature Versione 4 per evitare problemi.

- Se utilizzi i log CloudTrail degli eventi (opzione consigliata), scopri Identificazione delle richieste Amazon S3 Signature versione 2 mediante CloudTrail come interrogare e identificare tali richieste.
- Se utilizzi log di accesso al server Amazon S3, consulta Identificazione delle richieste di Signature versione 2 tramite i log degli accessi ad Amazon S3

Argomenti

Autenticazione delle richieste con l'utilizzo dell'API REST

Quando accedi ad Amazon S3 tramite REST, devi fornire gli elementi seguenti nella richiesta, in modo che questa possa essere autenticata:

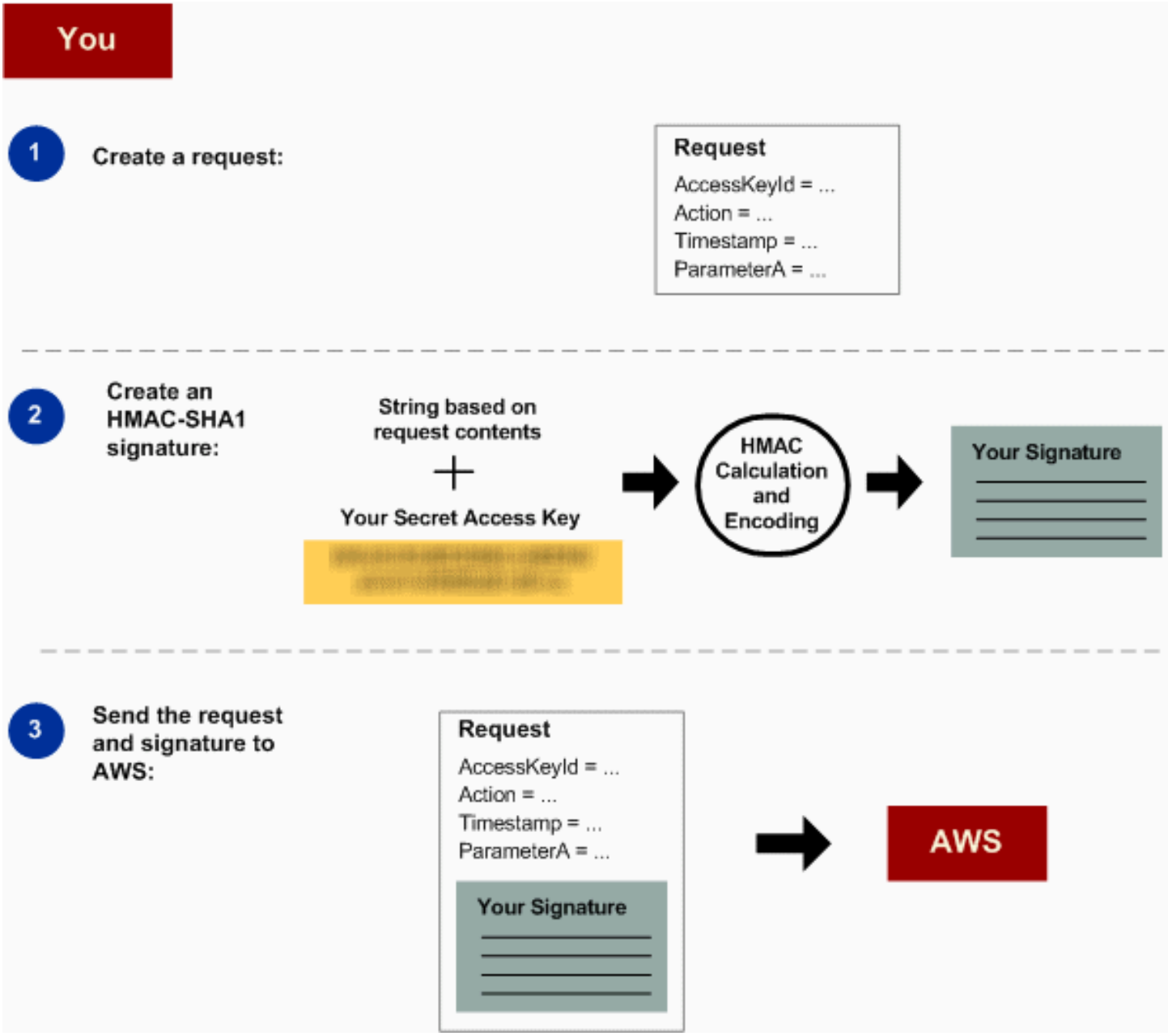
Elementi della richiesta

- AWS ID della chiave di accesso: ogni richiesta deve contenere l'ID della chiave di accesso dell'identità che stai utilizzando per inviare la richiesta.
- Firma: ogni richiesta deve contenere una firma di richiesta valida o la richiesta viene rifiutata.

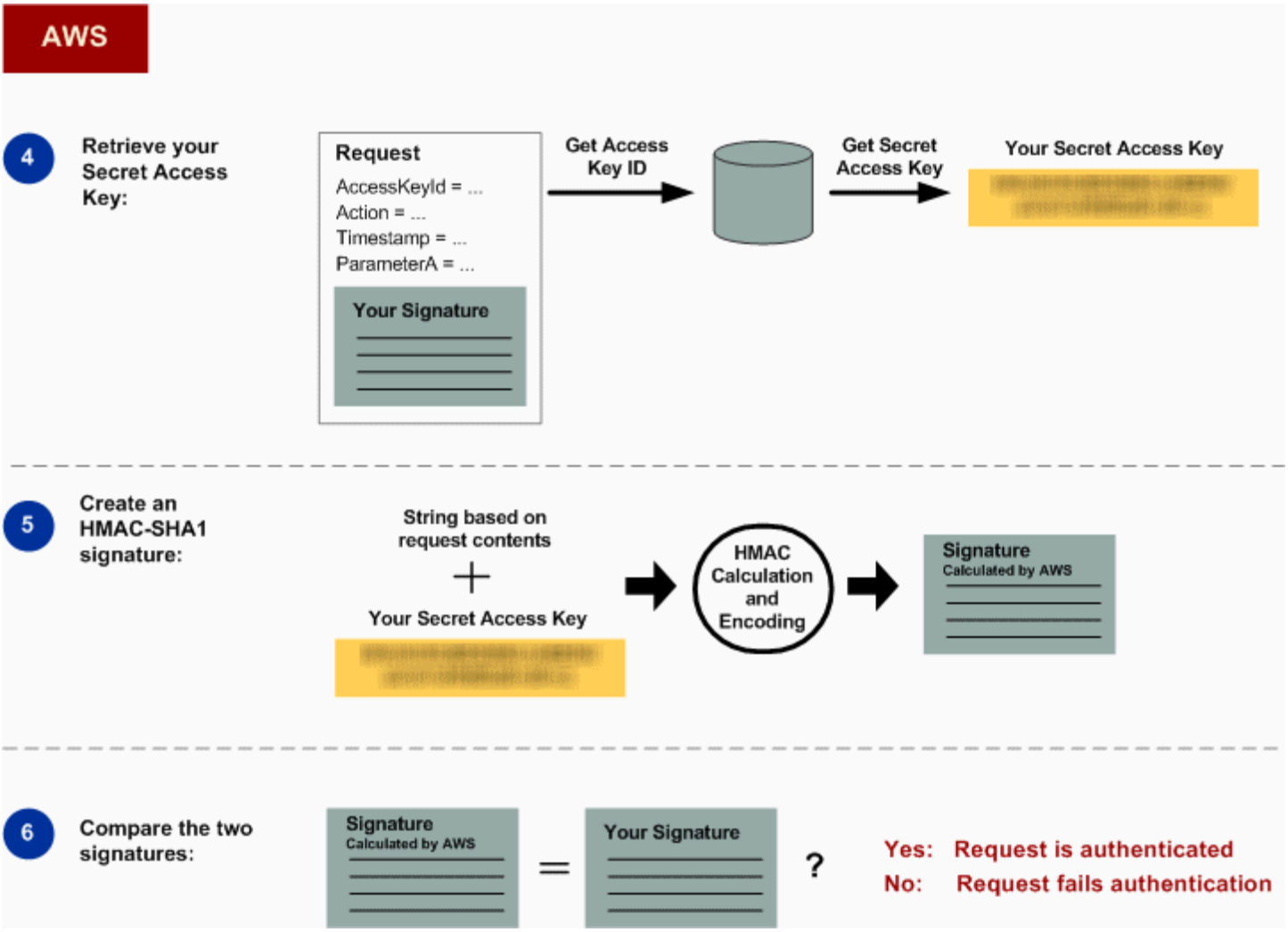
  Una firma di richiesta viene calcolata utilizzando la chiave di accesso segreta dell'interessato, che è un segreto condiviso unicamente tra quest'ultimo e AWS.
- Timestamp: ogni richiesta deve contenere la data e l'ora in cui è stata creata, sotto forma di stringa in UTC.
- Data: ogni richiesta deve contenere il timestamp.

  In funzione dell'operazione API in uso, è possibile fornire un'ora e una data di scadenza per la richiesta al posto del time stamp o in aggiunta a esso. consulta l'argomento sull'autenticazione per l'operazione specifica da eseguire per determinare ciò che è necessario.

Di seguito vengono presentate le fasi generali per l'autenticazione delle richieste in Amazon S3. Si presume che l'interessato disponga degli elementi necessari: credenziali di sicurezza, ID chiave di accesso e Secret Access Key.



1	Crea una richiesta a. AWS
2	Calcolare la firma utilizzando la Secret Access Key.
3	Inviare la richiesta ad Amazon S3. Includere l'ID chiave di accesso e la firma nella richiesta. Amazon S3 completa le tre fasi successive.



4	Amazon S3 utilizza l'ID chiave di accesso per cercare la chiave di accesso segreta.
5	Amazon S3 calcola una firma dai dati della richiesta e dalla chiave di accesso segreta tramite lo stesso algoritmo che hai utilizzato per calcolare la firma inviata nella richiesta .
6	Se la firma generata da Amazon S3 corrisponde a quella inviata nella richiesta, la richiesta viene considerata autentica. Se il confronto non ha esito positivo, la richiesta viene ignorata e Amazon S3 restituisce una risposta di errore.

Informazioni dettagliate sull'autenticazione

Per le informazioni dettagliate sull'autenticazione REST, consulta Firma e autenticazione delle richieste REST.

Firma e autenticazione delle richieste REST

Argomenti

> **Note**
>
> In questo argomento viene illustrato come autenticare le richieste tramite Signature Version 2. Amazon S3 supporta ora il più recente Signature Version 4. L'ultima versione è ora supportata in tutte le regioni e le regioni create dopo il 30 gennaio 2014 supporteranno solo Signature Version 4. Per ulteriori informazioni, consulta la sezione Autenticazione delle richieste (AWS Signature Version 4) nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Per autenticazione si intende il processo di attestazione dell'identità in un sistema. L'identità è un importante fattore per le decisioni di controllo degli accessi in Amazon S3. Le richieste vengono accettate o rifiutate in parte anche in base all'identità del richiedente. Ad esempio, il diritto di creare bucket è riservato agli sviluppatori registrati e, di default, è solo il proprietario del bucket in questione ad avere il diritto di creare oggetti al suo interno. Poiché gli sviluppatori inviano richieste che invocano questi privilegi, dovranno provare la propria identità al sistema autenticando le richieste: in questa sezione viene illustrato come procedere.

> **Note**
>
> I contenuti di questa sezione non si applicano alle richieste HTTP POST. Per ulteriori informazioni, consulta Caricamenti basati su browser tramite POST (versione 2 AWS della firma).

Per l'autenticazione, l'API REST di Amazon S3 utilizza uno schema HTTP personalizzato basato su un codice HMAC (Hash Message Authentication Code) con chiave. Per autenticare una richiesta, è necessario in primo luogo concatenare alcuni elementi della richiesta per formare una stringa, Per calcolare il codice HMAC di quella stringa si utilizza la chiave di accesso segreta AWS. Questo processo viene definito in modo informale "firma della richiesta", mentre il risultato dell'algoritmo HMAC viene definito firma perché simula le proprietà di sicurezza di una firma vera e propria. Infine, la firma viene aggiunta come parametro della richiesta tramite la sintassi descritta in questa sezione.

Quando il sistema riceve una richiesta autenticata, recupera la chiave di accesso segreta AWS dichiarata e la utilizza nello stesso modo per calcolare una firma per il messaggio ricevuto, quindi confronta la firma calcolata con quella presentata dal richiedente. Se le due firme coincidono, il sistema conclude che il richiedente ha accesso alla chiave di accesso segreta AWS e agisce quindi con l'autorità dell'entità principale a cui è stata assegnata la chiave. Se le due firme non coincidono, la richiesta viene scartata e il sistema risponde con un messaggio di errore.

Example Richiesta REST Amazon S3 autenticata

```
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=
```

Utilizzo di credenziali di sicurezza temporanee

Se la richiesta viene firmata tramite credenziali di sicurezza temporanee (consulta Esecuzione di richieste), è necessario includere nella richiesta il corrispondente token di sicurezza aggiungendo l'intestazione `x-amz-security-token`.

Quando si ottengono credenziali di sicurezza temporanee utilizzando l'API AWS Security Token Service, la risposta include le credenziali di sicurezza temporanee e un token di sessione. Il valore

del token di sessione deve essere fornito nell'intestazione `x-amz-security-token` durante l'invio di richieste ad Amazon S3. Per informazioni sull'API AWS Security Token Service fornita da IAM, consulta la sezione Operazioni nella Guida di riferimento alle API AWS Security Token Service.

Intestazione di autenticazione

L'API REST di Amazon S3 utilizza l'intestazione standard HTTP `Authorization` per passare le informazioni di autenticazione (il nome dell'intestazione standard è poco felice perché in realtà comunica informazioni sull'autenticazione, non sull'autorizzazione). Nell'ambito dello schema di autenticazione Amazon S3, l'intestazione Authorization ha il formato seguente:

```
Authorization: AWS AWSAccessKeyId:Signature
```

Al momento della registrazione, gli sviluppatori ricevono un ID chiave di accesso AWS e una chiave di accesso segreta AWS. Per l'autenticazione delle richieste, l'elemento `AWSAccessKeyId` identifica l'ID chiave di accesso utilizzata per calcolare la firma e, indirettamente, lo sviluppatore che invia la richiesta.

L'elemento `Signature` corrisponde al codice HMAC-SHA1 RFC 2104 di alcuni elementi della richiesta, quindi la parte `Signature` dell'intestazione Authorization varierà da richiesta a richiesta. Se la firma calcolata dal sistema corrisponde all'elemento `Signature` incluso nella richiesta, il richiedente avrà dimostrato di possedere la chiave di accesso segreta AWS. La richiesta verrà quindi elaborata con l'identità e l'autorità dello sviluppatore titolare della chiave.

Di seguito è riportato un esempio di grammatica che illustra la costruzione dell'intestazione `Authorization` per la richiesta. (Nell'esempio, l'elemento `\n` rappresenta il punto di codice Unicode `U+000A`, comunemente chiamato newline).

```
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 è un algoritmo definito in base allo standard RFC 2104 - Keyed-Hashing for Message Authentication. L'algoritmo prende come input due stringhe in bit, una chiave e un messaggio. Per l'autenticazione delle richieste in Amazon S3, utilizza la chiave di accesso segreta AWS (`YourSecretAccessKey`) come chiave e la codifica UTF-8 di `StringToSign` come messaggio. L'algoritmo HMAC-SHA1 restituirà sempre una stringa in byte, detta digest. Il parametro di richiesta `Signature` viene costruito da Base64 codificando il digest.

Standardizzazione della richiesta per la firma

È importante ricordare che, quando il sistema riceve una richiesta di autenticazione, confronta la firma calcolata con la firma fornita nella richiesta in `StringToSign`. Per questo motivo, devi calcolare la firma con lo stesso metodo utilizzato da Amazon S3. Il processo di stesura di una richiesta in una forma concordata per la firma viene definito standardizzazione.

Costruire l'elemento CanonicalizedResource

`CanonicalizedResource` rappresenta la risorsa Amazon S3 cui è destinata la richiesta. È possibile costruire l'elemento per una richiesta REST nel modo seguente:

Processo di lancio

1	Iniziare con una stringa vuota ("").
2	Se la richiesta specifica un bucket tramite l'intestazione Host HTTP (in stile hosting virtuale) , aggiungere il nome del bucket preceduto da "/" (ad es., "/nomebucket"). Per le richieste in stile percorso o senza un bucket specifico, non è necessaria alcuna azione. Per ulteriori informazioni sulle richieste in stile hosting virtuale, consulta Hosting virtuale dei bucket. Per una richiesta in stile hosting virtuale "https://awsexamplebucket1.s3.us-west-1.amazonaws.com/photos/puppy.jpg", l'elemento `CanonicalizedResource` è "/awsexamplebucket1". Per la richiesta in stile percorso, "https://s3.us-west-1.amazonaws.com/awsexamplebucket1/photos/puppy.jpg", l'elemento `CanonicalizedResource` è "".

3	Aggiungere la parte di percorso relativa all'URI della richiesta HTTP non decodificato, fino alla stringa di query (senza includerla). Per una richiesta in stile hosting virtuale "https://awsexamplebucket1.s3.us-west-1.amazonaws.com/photos/puppy.jpg", l'elemento `CanonicalizedResource` è "/awsexamplebucket1/photos/puppy.jpg". Per una richiesta di stile percorso, "https://s3.us-west-1.amazonaws.com/awsexamplebucket1/photos/puppy.jpg", `CanonicalizedResource` è "/awsexamplebucket1/photos/puppy.jpg". A questo punto, l'elemento `CanonicalizedResource` è lo stesso sia per le richieste in stile hosting virtuale che in stile percorso. Per una richiesta non indirizzata a un bucket, come GET Service, aggiungere "/".
4	Se la richiesta è indirizzata a una risorsa secondaria, come `?versioning` , `?location` , `?acl`, `?lifecycle` o `?versionid` , aggiungere la risorsa secondaria, il suo valore (se disponibile) e il punto interrogativo. Nel caso di più risorse secondarie, queste devono essere disposte in ordine lessicografico in base al nome della risorsa secondaria e separate da "&", ad es., ?acl&versionId=*valore*. Le sottorisorse che devono essere incluse durante la creazione dell' CanonicalizedResource elemento sono acl, lifecycle, location, logging, notification, partNumber, policy, RequestPayment, UploadId, uploads, versionId, versionID, versioning, versions e website. Se la richiesta specifica parametri della stringa di query che sostituiscono i valori dell'interstazione della risposta (consultare Get Object), aggiungere i parametri della stringa di query e i rispettivi valori. Al momento della firma questi valori non vanno codificati, ma i valori dei parametri devono essere codificati quando si invia la richiesta. I parametri della stringa di query in una richiesta GET includono `response-content-type` , `response-content-language` , `response-expires` , `response-cache-control` , `response-content-disposition` e `response-content-encoding` . Il parametro della stringa di `delete` query deve essere incluso quando si crea la richiesta Delete per più oggetti. CanonicalizedResource

Gli elementi di CanonicalizedResource che provengono dall'URI della richiesta HTTP devono essere firmati letteralmente così come appaiono nella richiesta HTTP, inclusi i metacaratteri per la codifica dell'URL.

L'elemento `CanonicalizedResource` può variare rispetto all'URI della richiesta HTTP. Nello specifico, se nella richiesta è utilizzata l'intestazione HTTP `Host` per la specifica di un bucket, quest'ultimo non compare nell'URI della richiesta HTTP. Tuttavia, nella `CanonicalizedResource` è sempre incluso il bucket. Nell'URI della richiesta possono comparire anche i parametri della stringa di query, che non sono comunque inclusi nella `CanonicalizedResource`. Per ulteriori informazioni, consulta Hosting virtuale dei bucket.

Costruire l'elemento CanonicalizedAmzHeaders

Per creare la CanonicalizedAmzHeaders parte di`StringToSign`, selezionate tutte le intestazioni di richiesta HTTP che iniziano con 'x-amz-' (utilizzando un confronto senza distinzione tra maiuscole e minuscole) e utilizzate il seguente processo.

CanonicalizedAmzHeaders processo

1	Convertire ciascun nome dell'intestazione HTTP in lettere minuscole. Ad esempio "`X-Amz-Date `" diventa "`x-amz-date `".
2	Disporre la raccolta delle intestazioni in ordine lessicografico in base al nome dell'inte stazione.
3	Combina i campi di intestazione con lo stesso nome in un'unica coppia «header-n ame:comma-separated-value-list» come prescritto da RFC 2616, sezione 4.2, senza spazi tra i valori. Ad esempio, le due intestazioni con metadata "`x-amz-meta-username: fred`" e "`x-amz-meta-username: barney `" verrebbero combinate nell'intestazione singola "`x-amz-meta-username: fred,barney `".
4	"Aprire" le intestazioni più lunghe che occupano più righe (come consentito dallo standard RFC 2616, sezione 4.2) sostituendo gli spazi di folding (newline compreso) con un singolo spazio.
5	Rimuovere gli spazi attorno ai due punti nell'intestazione. Ad esempio, l'intestazione "`x-amz-meta-username: fred,barney `" diventerebbe "`x-amz-meta-usernam e:fred,barney `"
6	Infine, aggiungere un carattere newline (`U+000A`) a ogni intestazione standardizzata nell'elenco risultante. Costruisci l' CanonicalizedResource elemento concatenando tutte le intestazioni di questo elenco in un'unica stringa.

Elementi di intestazione HTTP posizionali e denominati StringToSign

I primi elementi dell'intestazione di `StringToSign` (Content-Type, Date e Content-MD5) sono posizionali per natura. `StringToSign` non include i nomi delle intestazioni, ma solo i valori tratti dalla richiesta. Al contrario, gli elementi "`x-amz-`" sono denominati. In sono inclusi sia i nomi, sia i valori dell'intestazion `StringToSign`.

Se un'intestazione posizionale richiamata per la definizione di `StringToSign` non è presente nella richiesta (ad esempio, `Content-Type` o `Content-MD5` sono facoltative per le richieste PUT e prive di senso per le richieste GET), la stringa vuota ("") per tale posizione va sostituita.

Necessità del time stamp

Per le richieste autenticate, è obbligatorio un time stamp valido (tramite intestazione `Date` HTTP o un'alternativa `x-amz-date`). Inoltre, alla ricezione di una richiesta autenticata, il timestamp del client incluso nella richiesta non deve differire di oltre 15 minuti rispetto all'orario di sistema in Amazon S3. In caso contrario, la richiesta avrà esito negativo con il codice di errore `RequestTimeTooSkewed`. Tali restrizioni hanno lo scopo di limitare le possibilità che una richiesta intercettata possa essere riproposta da un sistema avversario. Per una maggiore protezione contro le intercettazioni, utilizzare il protocollo di trasferimento HTTPS per le richieste autenticate.

> **Note**
>
> Il vincolo relativo alla convalida della data della richiesta è valido solo per le richieste autenticate che non utilizzano l'autenticazione stringa di query. Per ulteriori informazioni, consulta Alternativa per l'autenticazione di una richiesta tramite stringa di query.

Alcune librerie client HTTP non danno la possibilità di impostare l'intestazione `Date` per una richiesta. Se includere il valore dell'intestazione "Date" nelle intestazioni standardizzate crea difficoltà, è possibile impostare il time stamp per la richiesta utilizzando l'intestazione "`x-amz-date`". Il valore dell'intestazione `x-amz-date` deve essere in uno dei formati RFC 2616 (http://www.ietf.org/rfc/rfc2616.txt). Quando in una richiesta è presente un'intestazione `x-amz-date`, durante il calcolo della firma della richiesta il sistema ignorerà qualsiasi intestazione di tipo `Date`. Di conseguenza, se si include l'intestazione `x-amz-date`, va utilizzata la stringa vuota per il valore `Date` durante la costruzione della stringa `StringToSign`. Nella prossima sezione sarà riportato un esempio.

Esempi di autenticazione

Gli esempi in questa sezione utilizzano le credenziali (non funzionanti) incluse nella seguente tabella.

Parametro	Valore
AWSAccessKeyId	`AKIAIOSFODNN7EXAMPLE`
AWSSecret AccessKey	`wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`

Negli elementi `StringToSign` di esempio, la formattazione non è significativa, mentre l'elemento `\n` rappresenta il punto di codice Unicode `U+000A`, comunemente chiamato newline. Inoltre, negli esempi viene utilizzato "+0000" per identificare il fuso orario. Allo stesso scopo è anche possibile utilizzare "GMT", ma le firme mostrate nell'esempio saranno diverse.

Richiesta GET di un oggetto

Nell'esempio viene richiesto un oggetto dal bucket awsexamplebucket1.

Richiesta	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`

Nota che CanonicalizedResource include il nome del bucket, ma l'URI della richiesta HTTP no. (Il bucket è specificato dall'intestazione Host).

> **Note**
>
> Il seguente script Python calcola la firma precedente utilizzando i parametri forniti. È possibile utilizzare questo script per creare le proprie firme, sostituendo le chiavi e se necessario. StringToSign

```
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()}")
```

Richiesta PUT di un oggetto

Nell'esempio viene inserito un oggetto nel bucket awsexamplebucket1.

Richiesta	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:` `iqRzw+ileNPu1fhspnRs8nOjjIA=`	`PUT\n` `\n` `image/jpeg\n` `Tue, 27 Mar 2007 21:15:45 +0000\n` `/awsexamplebucket1/photos/puppy.jpg`

Nota l'intestazione Content-Type nella richiesta e in. StringToSign Si noti inoltre che l'oggetto Content-MD5 viene lasciato vuoto in StringToSign, poiché non è presente nella richiesta.

Elenco

Nell'esempio viene recuperato l'elenco dei contenuti del bucket awsexamplebucket1.

Richiesta	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/`

Notate la barra finale CanonicalizedResource e l'assenza dei parametri della stringa di query.

Recupero

Nell'esempio viene recuperata la risorsa secondaria relativa alla policy di controllo degli accessi per il bucket "awsexamplebucket1".

Richiesta	StringToSign
`GET /?acl HTTP/1.1` `Host: awsexamplebucket1.s3.us-west-1.amazonaws.com` `Date: Tue, 27 Mar 2007 19:44:46 +0000` `Authorization: AWS AKIAIOSFODNN7EXAMPLE:82ZHiFIjc+WbcwFKGUVEQspPn+0=`	`GET\n` `\n` `\n` `Tue, 27 Mar 2007 19:44:46 +0000\n` `/awsexamplebucket1/?acl`

Notate come il parametro della stringa di query della subresource è incluso in. CanonicalizedResource

Eliminazione

In questo esempio viene eliminato un oggetto dal bucket "awsexamplebucket1" utilizzando lo stile percorso e l'alternativa Date.

Richiesta	StringToSign
`DELETE /awsexamplebucket1/photos/puppy.jpg HTTP/1.1` `User-Agent: dotnet` `Host: s3.us-west-1.amazonaws.com` `Date: Tue, 27 Mar 2007 21:20:27 +0000` `x-amz-date: Tue, 27 Mar 2007 21:20:26 +0000` *`Authorization: AWS AKIAIOSFODNN7EXAMPLE:XbyTlbQdu9Xw5o8P4iMwPktxQd8=`*	`DELETE\n` `\n` `\n` `Tue, 27 Mar 2007 21:20:26 +0000\n` `/awsexamplebucket1/photos/puppy.jpg`

Nota come abbiamo usato il metodo alternativo 'x-amz-date' per specificare la data (perché la nostra libreria client ci ha impedito di impostare la data, ad esempio). In questo caso, il metodo `x-amz-date` prevale sull'intestazione `Date`. Di conseguenza, la voce relativa alla data nella firma deve contenere il valore dell'intestazione `x-amz-date`.

Caricamento

In questo esempio viene caricato un oggetto in un bucket CNAME in stile hosting virtuale con metadata.

Richiesta	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`	`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:crc32\n`

Richiesta	StringToSign
`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=`*	`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`

Osserva il modo in cui le intestazioni "x-amz-" vengono ordinate, private degli spazi aggiuntivi e convertite in caratteri minuscoli. Inoltre, più intestazioni con lo stesso nome sono state unite, con i valori separati da virgole.

Si noti come solo le intestazioni delle entità HTTP `Content-Type` e `Content-MD5` siano presenti in `StringToSign`, mentre le intestazioni dell'altra entità `Content-*` non lo sono.

Si noti nuovamente che il nome del bucket è presente in `CanonicalizedResource`, ma non nell'URI della richiesta HTTP. (Il bucket è specificato dall'intestazione Host).

Elenco di tutti i bucket personali

Richiesta	StringToSign
`GET / HTTP/1.1` `Host: s3.us-west-1.amazonaws.com` `Date: Wed, 28 Mar 2007 01:29:59 +0000` *`Authorization: AWS AKIAIOSFODNN7EXAMPLE:qGdzdE RIC03wnaRNKh6OqZehG9s=`*	`GET\n` `\n` `\n` `Wed, 28 Mar 2007 01:29:59 +0000\n` `/`

Chiavi Unicode

Richiesta	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**
>
> Gli elementi inclusi in `StringToSign` derivanti dall'URI della richiesta sono stati presi letteralmente, comprensivi di codifica dell'URL e uso delle maiuscole.

Problemi con la firma delle richieste REST

In caso di errore nell'autenticazione delle richieste REST, il sistema risponde alla richiesta con un documento di errore XML, che contiene informazioni concepite per aiutare gli sviluppatori a individuare il problema. In particolare, l'elemento `StringToSign` del documento di errore `SignatureDoesNotMatch` indica esplicitamente il tipo di standardizzazione della richiesta usata dal sistema.

Alcuni kit di strumenti inseriscono tacitamente intestazioni non note in precedenza, ad esempio nel caso dell'aggiunta dell'intestazione `Content-Type` durante un'operazione PUT. In gran parte dei casi, il valore dell'intestazione inserita rimane costante: questo consente di scoprire le intestazioni mancanti utilizzando strumenti come Ethereal o tcpmon.

Alternativa per l'autenticazione di una richiesta tramite stringa di query

È possibile autenticare alcuni tipi di richieste passando le informazioni necessarie come parametri di una stringa di query, invece di utilizzare l'intestazione HTTP `Authorization`. Questa soluzione è utile per abilitare l'accesso diretto da parte di browser di terze parti ai dati Amazon S3 privati senza proxy della richiesta. L'idea è costruire di una richiesta "prefirmata" e codificarla come URL recuperabile da parte del browser di un utente finale. È inoltre possibile limitare una richiesta prefirmata specificando un periodo di scadenza.

Per ulteriori informazioni sull'utilizzo di parametri di query per autenticare le richieste, consulta la sezione Autenticazione delle richieste: Uso dei parametri di query (AWS Signature Version 4) nella Documentazione di riferimento delle API di Amazon Simple Storage Service. Per alcuni esempi di utilizzo degli SDK AWS per la generazione di URL prefirmati, consulta Condivisione di oggetti mediante URL prefirmati.

Creazione di una firma

Di seguito è riportato un esempio di richiesta REST Amazon S3 autenticata tramite stringa di query.

```
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
```

Il metodo di autenticazione delle richieste tramite stringa di query non richiede particolari intestazioni HTTP. Gli elementi necessari all'autenticazione sono invece specificati come parametri della stringa di query:

Nome parametro stringa di query	Valore di esempio	Descrizione
`AWSAccessKeyId`	`AKIAIOSFODNN7EXAMPLE`	L'ID chiave di accesso AWS. Specifica la chiave di accesso segreta AWS utilizzata per firmare la richiesta e, indirettamente, l'identità dello sviluppatore che invia la richiesta.
`Expires`	`1141889120`	L'ora di scadenza della firma, specificata come numero di secondi dal valore epoca (Unix epoch, ovvero 00:00:00 UTC del 1° gennaio 1970). Se ricevuta dopo l'ora indicata (secondo il server), la richiesta verrà rifiutata.

Nome parametro stringa di query	Valore di esempio	Descrizione
`Signature`	`vjbyPxybdZaNmGa%2B yT272YEAiv4%3D`	La codifica URL della codifica Base64 dell'HMAC-SHA1 di. StringToSign

Il metodo di autenticazione delle richieste tramite stringa di query varia leggermente rispetto al metodo tradizionale, ma solo nel formato del parametro di richiesta `Signature` e nell'elemento `StringToSign`. Di seguito è riportato un esempio di grammatica che illustra il metodo di autenticazione delle richieste tramite stringa di query.

```
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` è l'ID chiave di accesso segreta AWS assegnata da Amazon al momento della registrazione come sviluppatore Amazon Web Service. Si noti come il parametro `Signature` sia codificato come URL per renderlo adatto all'inserimento nella stringa di query. Inoltre, in `StringToSign`, l'elemento posizionale HTTP `Date` è stato sostituito con l'elemento `Expires`, mentre `CanonicalizedAmzHeaders` e `CanonicalizedResource` restano invariati.

> **Note**
>
> Nel metodo di autenticazione stringa di query, non si utilizzano né l'elemento `Date` né l'intestazione `x-amz-date request` per il calcolo della stringa da firmare.

Autenticazione di una richiesta tramite stringa di query

Richiesta	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`

Si presuppone che, al momento di inviare la richiesta GET, il browser non fornisca le intestazioni Content-MD5 o Content-Type e non imposti intestazioni x-amz-, quindi queste parti sono lasciate vuote nell'elemento `StringToSign`.

Utilizzo della codifica Base64

Le firme di richiesta HMAC devono essere codificate in formato Base64. La codifica Base64 converte la firma in una stringa ASCII semplice che è possibile allegare alla richiesta. Se utilizzati in un URI, i caratteri che possono essere presenti in una firma, ad esempio il più (+), la barra (/) e l'uguale (=), devono essere codificati. Ad esempio, se il codice di autenticazione include un simbolo più (+), nella richiesta va codificato come %2B. La codifica per la barra è %2F, mentre per l'uguale è %3D.

Per esempi di codifica Base64, consulta di Amazon 3 Esempi di autenticazione.

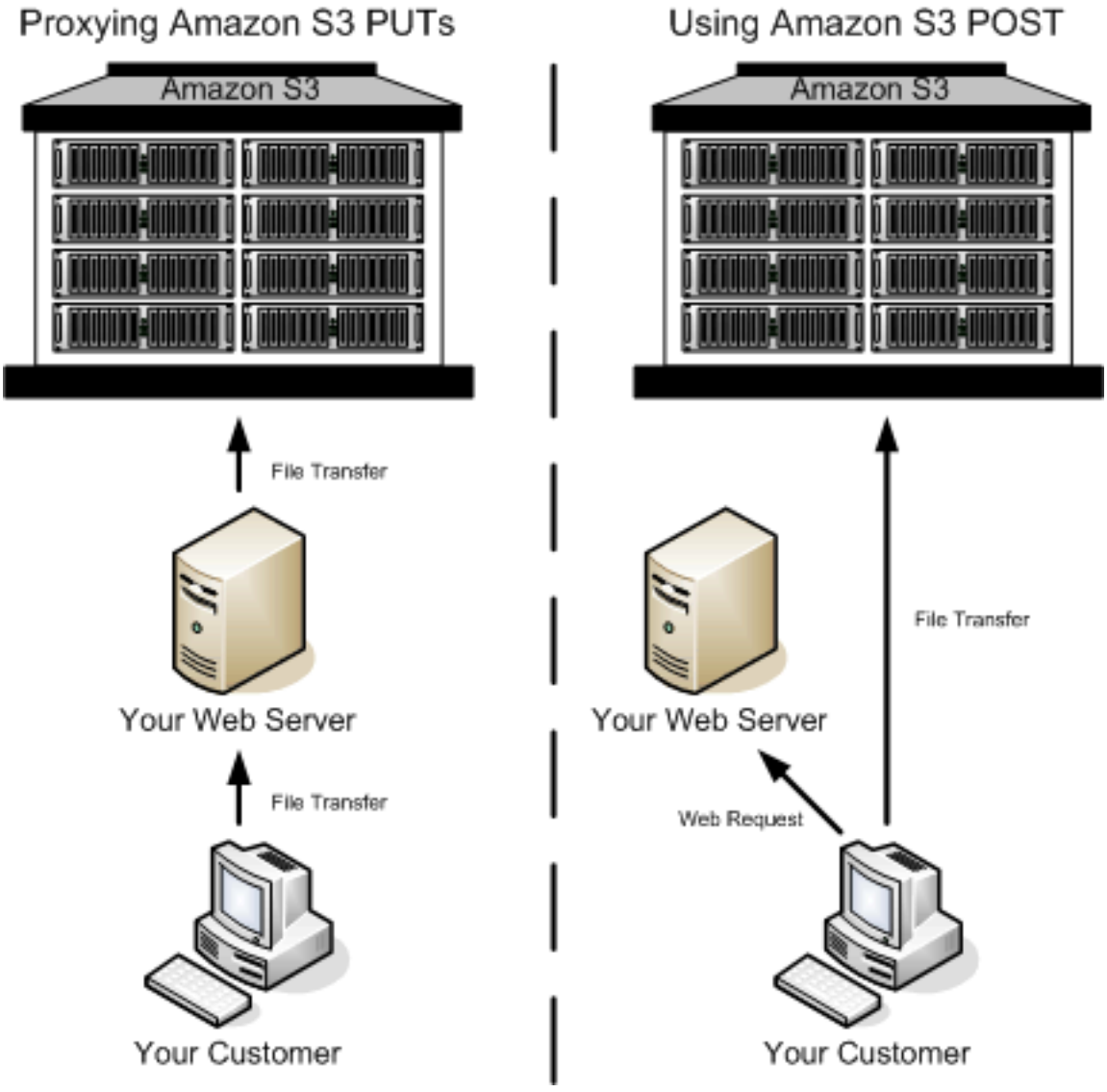
Caricamenti basati su browser tramite POST (versione 2 AWS della firma)

Amazon S3 supporta POST, che permette agli utenti di caricare contenuti direttamente in Amazon S3. POST è progettato per semplificare i caricamenti, ridurne la latenza e risparmiare denaro sulle applicazioni in cui gli utenti caricano dati da archiviare in Amazon S3.

> **Note**
>
> L'autenticazione delle richieste discussa in questa sezione si basa sulla versione 2 di AWS Signature, un protocollo per l'autenticazione delle richieste API in entrata ai servizi. AWS Amazon S3 ora supporta complessivamente la versione 4 di Signature, un protocollo per l'autenticazione delle richieste API in entrata ai AWS servizi. Regioni AWS Al momento,

Regioni AWS creato prima del 30 gennaio 2014, continuerà a supportare il protocollo precedente, la versione 2 di Signature. Le regioni create dopo il 30 gennaio 2014 supporteranno solo Signature Version 4 e pertanto tutte le richieste a tali regioni devono essere effettuate con Signature Version 4. Per ulteriori informazioni, consulta Autenticazione delle richieste nei caricamenti basati su browser utilizzando POST (AWS Signature versione 4) nel riferimento all'API di Amazon Simple Storage Service.

La figura seguente mostra un caricamento con POST in Amazon S3.



Caricamento tramite POST

1	L'utente apre un browser Web e accede alla tua pagina Web.

2	La tua pagina Web contiene un modulo HTTP con tutte le informazioni necessarie all'utente per caricare contenuti in Amazon S3.
3	L'utente carica contenuti direttamente in Amazon S3.

> **Note**
>
> L'autenticazione stringa di query non è supportata per POST.

Moduli HTML (versione 2 AWS della firma)

Argomenti

- Codifica dei moduli HTML
- Dichiarazione del modulo HTML
- Campi del modulo HTML
- Costruzione della policy
- Costruzione di una firma
- Reindirizzamento

Quando comunichi con Amazon S3, utilizzi in genere l'API REST o SOAP per eseguire operazioni di inserimento, recupero e di altro tipo. Con POST, gli utenti caricano i dati direttamente in Amazon S3 tramite i propri browser, che non possono elaborare l'API SOAP o creare una richiesta REST PUT.

> **Note**
>
> Il supporto di SOAP su HTTP non viene più utilizzato ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non sono supportate per SOAP. Invece di usare SOAP, ti consigliamo di utilizzare l'API REST o gli AWS SDK.

Per permettere agli utenti di caricare contenuti in Amazon S3 utilizzando i propri browser, devi utilizzare moduli HTML. I moduli HTML si compongono di una dichiarazione del modulo e di campi del modulo. La dichiarazione del modulo contiene informazioni generali sulla richiesta. I campi del

modulo contengono informazioni dettagliate sulla richiesta, nonché la policy utilizzata per autenticarla e garantire che soddisfi le condizioni specificate.

> **Note**
>
> I dati e i limiti del modulo (esclusi i contenuti del file) non possono superare 20 KB.

Questa sezione spiega come utilizzare i moduli HTML.

Codifica dei moduli HTML

Il modulo e la policy devono avere la codifica UTF-8. È possibile applicare la codifica UTF-8 al modulo specificandola nell'intestazione HTML o come intestazione di una richiesta.

> **Note**
>
> La dichiarazione del modulo HTML non accetta parametri di autenticazione stringa di query.

Di seguito è illustrato un esempio della codifica UTF-8 nell'intestazione HTML:

```
<html>
  <head>
    ...
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    ...
  </head>
  <body>
```

Di seguito è illustrato un esempio della codifica UTF-8 nell'intestazione di una richiesta:

```
Content-Type: text/html; charset=UTF-8
```

Dichiarazione del modulo HTML

La dichiarazione del modulo presenta tre componenti: l'operazione, il metodo e il tipo di inquadramento. Se uno di questi valori non è impostato correttamente, la richiesta ha esito negativo.

L'operazione specifica l'URL che elabora la richiesta, il quale deve essere impostato sull'URL del bucket. Ad esempio, se il nome del bucket è `awsexamplebucket1` e la regione è Stati Uniti occidentali (California settentrionale), l'URL è `https://awsexamplebucket1.s3.us-west-1.amazonaws.com/`.

> **Note**
>
> Il nome della chiave è specificato nel campo del modulo.

Il metodo deve essere POST.

Il tipo di inquadramento (enctype) deve essere specificato e impostato su multipart/form-data sia per i caricamenti di file sia per i caricamenti di aree di testo. Per ulteriori informazioni, consulta RFC 1867.

Example

L'esempio seguente è la dichiarazione di un modulo per il bucket "awsexamplebucket".

```
<form action="https://awsexamplebucket1.s3.us-west-1.amazonaws.com/" method="post"

enctype="multipart/form-data">
```

Campi del modulo HTML

Nella seguente tabella sono descritti i campi che possono essere utilizzati in un modulo HTML.

> **Note**
>
> La variabile `${filename}` viene sostituita automaticamente con il nome del file fornito dall'utente ed è riconosciuta da tutti i campi del modulo. Se il browser o il client fornisce un percorso completo o parziale al file, verrà utilizzato solo il testo che segue l'ultima barra (/) o barra rovesciata (\). Ad esempio, "C:\Program Files\directory1\file.txt" viene interpretato come "file.txt". Se non vengono forniti file o nomi di file, la variabile viene sostituita con una stringa vuota.

Nome campo	Descrizione	Richiesto
`AWSAccessKeyId`		

Nome campo	Descrizione	Richiesto
	L'ID della chiave di AWS accesso del proprietario del bucket che concede a un utente anonimo l'accesso per una richiesta che soddisfa l'insieme di vincoli della policy. Questo campo è obbligatorio se la richiesta include un documento di policy.	Condizionale
`acl`	Lista di controllo accessi (ACL) Amazon S3. Se si specifica una lista di controllo accessi non valida, viene generato un errore. Per ulteriori informazioni sulle ACL, consulta Liste di controllo degli accessi (ACL). Tipo: string Default: private Valori validi: `private | public-read | public-read-write | aws-exec-read | authenticated-read | bucket-owner-read | bucket-owner-full-control`	No
`Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires`	Intestazioni specifiche per REST. Per ulteriori informazioni, consulta PUT Object.	No

Nome campo	Descrizione	Richiesto
`key`	Il nome della chiave caricata. Per utilizzare il nome del file fornito dall'utente, utilizzare la variabile ${filename}. Ad esempio, se l'utente Betty carica il file lolcatz.jpg e si specifica /user/betty/${filename}, il file viene archiviato come /user/betty/lolcatz.jpg. Per ulteriori informazioni, consulta Utilizzo dei metadati degli oggetti.	Sì
`policy`	Policy di sicurezza che descrive cosa è permesso nella richiesta. Le richieste senza policy di sicurezza sono considerate anonime e hanno esisto positivo solo su bucket pubblicamente scrivibili.	No

Nome campo	Descrizione	Richiesto
`success_action_redirect, redirect`	L'URL a cui viene reindirizzato il client dopo il corretto caricamento. Amazon S3 aggiunge i valori di bucket, chiave ed ETag come parametri della stringa di query all'URL. Se success_action_redirect non è specificato, Amazon S3 restituisce il tipo di documento vuoto specificato nel campo success_action_status. Se Amazon S3 non riesce a interpretare l'URL, ignora il campo . Se il caricamento non riesce, Amazon S3 visualizza un errore e non reindirizza l'utente a un URL. Per ulteriori informazioni, consulta Reindirizzamento. **Note** Il nome del campo di reindirizzamento è obsoleto e in futuro il supporto del nome del campo di reindirizzamento verrà eliminato.	No

Nome campo	Descrizione	Richiesto
`success_action_status`	Il codice di stato restituito al client dopo il corretto caricamento, se non è specificato success_action_redirect. I valori validi sono 200, 201 o 204 (default). Se il valore è impostato su 200 o 204, Amazon S3 restituisce un documento vuoto con codice di stato 200 o 204. Se il valore è impostato su 201, Amazon S3 restituisce un documento XML con codice di stato 201. Per informazioni sul contenuto del documento XML, consulta POST Object. Se il valore non è impostato o non è valido, Amazon S3 restituisce un documento vuoto con codice di stato 204. **Note** Alcune versioni di Adobe Flash Player non gestiscono correttamente le risposte HTTP con corpo vuoto. Per supportare i caricamenti tramite Adobe Flash, si consiglia di impostare `success_action_status` su 201.	No

Nome campo	Descrizione	Richiesto
`signature`	La firma HMAC costruita utilizzando la chiave di accesso segreta che corrisponde a quella fornita. AWSAccessKeyId Questo campo è obbligatorio se la richiesta include un documento di policy. Per ulteriori informazioni, consulta Identity and Access Management per Amazon S3.	Condizionale
`x-amz-security-token`	Token di sicurezza utilizzato dalle credenziali per la sessione Se la richiesta utilizza Amazon DevPay , sono necessari due campi del `x-amz-security-token` modulo: uno per il token del prodotto e uno per il token utente. Se la richiesta utilizza le credenziali per la sessione, richiede un modulo `x-amz-security-token` . Per ulteriori informazioni, consulta Credenziali di sicurezza temporanee nella Guida per l'utente di IAM.	No
Altri nomi di campo con x-amz-meta il prefisso -	Metadata specificati dall'utente. Amazon S3 non convalida o utilizza questi dati. Per ulteriori informazioni, consulta PutObject.	No

Nome campo	Descrizione	Richiesto
file	File o contenuto di testo. Il file o il contenuto deve essere l'ultimo campo del modulo. Tutti i campi al di sotto di questo vengono ignorati. Non è possibile caricare più di un file alla volta.	Sì

Costruzione della policy

Argomenti

- Expiration
- Condizioni
- Corrispondenza delle condizioni
- Utilizzo di escape con caratteri

La policy è un documento JSON con codifica UTF-8 e Base64 che specifica le condizioni che la richiesta deve soddisfare ed è utilizzata per autenticare il contenuto. A seconda di come si progettano i documenti di policy, è possibile utilizzarli per caricamento, per utente, per tutti i caricamenti o secondo altre progettazioni in base alle esigenze.

> **Note**
>
> Sebbene il documento di policy sia facoltativo, è fortemente consigliato rispetto a rendere il bucket pubblicamente scrivibile.

Di seguito è illustrato un esempio di documento di policy:

```
{ "expiration": "2007-12-01T12:00:00.000Z",

  "conditions": [
```

```
    {"acl": "public-read" },

    {"bucket": "awsexamplebucket1" },

    ["starts-with", "$key", "user/eric/"],

  ]

}
```

Il documento di policy contiene la scadenza e le condizioni.

Expiration

L'elemento scadenza specifica la data di scadenza della policy in formato ISO 8601 UTC. Ad esempio, "2007-12-01T12:00:00.000Z" specifica che la policy non è valida dopo mezzanotte, orario UTC, del 01/12/2007. In una policy la scadenza è obbligatoria.

Condizioni

Le condizioni nel documento di policy convalidano i contenuti dell'oggetto caricato. Ogni campo del modulo specificato nel modulo (ad eccezione di firmeAWSAccessKeyId, file, criteri e nomi di campo che hanno un prefisso x-ignore-) deve essere incluso nell'elenco delle condizioni.

> **Note**
>
> Se si dispone di più campi con lo stesso nome, i valori devono essere separati da virgole. Ad esempio, se avete due campi denominati "x-amz-meta-tag" e il primo ha il valore «Ninja» e il secondo ha il valore «Stallman», il documento di policy deve essere impostato su. `Ninja,Stallman`
> Tutte le variabili all'interno del modulo vengono estese prima della convalida della policy. Pertanto, tutte le corrispondenze delle condizioni devono essere eseguite rispetto ai campi estesi. Ad esempio, se si imposta il campo della chiave su `user/betty/${filename}`, la policy potrebbe essere `[ "starts-with", "$key", "user/betty/" ]`. Non inserire `[ "starts-with", "$key", "user/betty/${filename}" ]`. Per ulteriori informazioni, consulta Corrispondenza delle condizioni.

La tabella seguente descrive le condizioni del documento di policy.

Nome elemento	Descrizione
acl	Specifica le condizioni che l'ACL deve soddisfare. Supporta la corrispondenza esatta e `starts-with` .
content-length-range	Specifica la dimensione minima e massima consentita per il contenuto caricato. Supporta la corrispondenza dell'intervallo.
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	Intestazioni specifiche per REST. Supporta la corrispondenza esatta e `starts-with` .
key	Il nome della chiave caricata. Supporta la corrispondenza esatta e `starts-with` .
success_action_redirect, redirect	L'URL a cui viene reindirizzato il client dopo il corretto caricamento. Supporta la corrispondenza esatta e `starts-with` .
success_action_status	Il codice di stato restituito al client dopo il corretto caricamento, se non è specificato success_action_redirect. Supporta la corrispondenza esatta.
x-amz-security-token	Token DevPay di sicurezza Amazon. Ogni richiesta che utilizza Amazon DevPay richiede due campi del `x-amz-security-token` modulo: uno per il token del prodotto e uno per il token utente. Di conseguenza, i valori devono essere separati da virgole. Ad esempio, se il token utente è `eW91dHViZQ==` e il token prodotto

Nome elemento	Descrizione
	è `b0hnNVNKWVJIQTA=` , si imposta la voce di policy su: `{ "x-amz-security-token": "eW91dHViZQ==,b0hnNVNKWVJIQTA=" }` .
Altri nomi di campo preceduti da x-amz-meta -	Metadata specificati dall'utente. Supporta la corrispondenza esatta e `starts-with` .

> **Note**
>
> Se il kit di strumenti aggiunge ulteriori campi (ad esempio, Flash aggiunge il nome del file), è necessario aggiungerli al documento di policy. Se puoi controllare questa funzionalità, aggiungi il prefisso `x-ignore-` al campo in modo che Amazon S3 ignori la funzionalità e non ne alteri le versioni future.

Corrispondenza delle condizioni

La tabella seguente descrive i tipi di corrispondenza delle condizioni. Sebbene occorra specificare una condizione per ciascun campo del modulo specificato nel modulo, è possibile creare criteri di corrispondenza più complessi specificando più condizioni per un campo del modulo.

Condition	Descrizione
Corrispondenze esatte	Le corrispondenze esatte verificano che i campi corrispondano a specifici valori. Questo esempio indica che l'ACL deve essere impostata su public-read: `{"acl": "public-read" }` Questo esempio rappresenta un modo alternativo per indicare che l'ACL deve essere impostata su public-read: `[ "eq", "$acl", "public-read" ]`

Condition	Descrizione
Inizia con	Se il valore deve iniziare con un determinato valore, utilizzare starts-with. Questo esempio indica che la chiave deve iniziare con user/betty: `["starts-with", "$key", "user/betty/"]`
Corrispondenza di qualsiasi contenuto	Per configurare la policy in modo da autorizzare qualsiasi contenuto in un campo, utilizzare starts-with con un valore vuoto. Questo esempio permette qualsiasi success_action_redirect: `["starts-with", "$success_action_redirect", ""]`
Specifica degli intervalli	Per i campi che accettano intervalli, separare gli intervalli superiori e inferiori con una virgola. Questo esempio permette una dimensione dei file da 1 a 10 megabyte: `["content-length-range", 1048579, 10485760]`

Utilizzo di escape con caratteri

La tabella seguente descrive i caratteri da inserire in una sequenza di escape in un documento di policy.

Sequenza di escape	Descrizione
\\	Barra rovesciata
\$	Simbolo del dollaro

Sequenza di escape	Descrizione
\b	Backspace
\f	Avanzamento modulo
\n	Nuova riga
\r	Ritorno a capo
\t	Tabulatore orizzontale
\v	Tabulatore verticale
\u*xxxx*	Tutti i caratteri Unicode

Costruzione di una firma

Fase	Descrizione
1	Codificare la policy utilizzando UTF-8.
2	Codificare i byte UTF-8 utilizzando Base64.
3	Firmare la policy con la Secret Access Key utilizzando HMAC SHA-1.
4	Codificare la firma SHA-1 utilizzando Base64.

Per ulteriori informazioni sull'autenticazione, consulta Identity and Access Management per Amazon S3.

Reindirizzamento

Questa sezione descrive come gestire i reindirizzamenti.

Reindirizzamento generico

Al completamento della richiesta POST, l'utente viene reindirizzato alla posizione specificata nel campo `success_action_redirect`. Se Amazon S3 non riesce a interpretare l'URL, ignora il campo `success_action_redirect`.

Se `success_action_redirect` non è specificato, Amazon S3 restituisce il tipo di documento vuoto specificato nel campo `success_action_status`.

Se la richiesta POST non riesce, Amazon S3 visualizza un errore e non fornisce un reindirizzamento.

Reindirizzamento precedente al caricamento

Se il bucket è stato creato utilizzando **<** CreateBucketConfiguration **>**, gli utenti finali potrebbero richiedere un reindirizzamento. In tal caso, alcuni browser potrebbero gestire il reindirizzamento in modo errato. Ciò è relativamente raro, ma vi è maggiore probabilità che si verifichi subito dopo la creazione del bucket.

Carica esempi (versione 2 AWS della firma)

Argomenti

- Caricamento di un file
- Caricamento di un'area di testo

> **Note**
>
> L'autenticazione delle richieste discussa in questa sezione si basa sulla versione 2 di AWS Signature, un protocollo per l'autenticazione delle richieste API in entrata ai servizi. AWS Amazon S3 ora supporta complessivamente la versione 4 di Signature, un protocollo per l'autenticazione delle richieste API in entrata ai AWS servizi. Regioni AWS Al momento, Regioni AWS creato prima del 30 gennaio 2014, continuerà a supportare il protocollo precedente, la versione 2 di Signature. Le regioni create dopo il 30 gennaio 2014 supporteranno solo Signature Version 4 e pertanto tutte le richieste a tali regioni devono essere effettuate con Signature Version 4. Per ulteriori informazioni, consulta Esempi:

caricamento basato su browser tramite HTTP POST (utilizzo della versione 4 di AWS Signature) nel riferimento all'API di Amazon Simple Storage Service.

Caricamento di un file

Questo esempio illustra il processo completo di costruzione di una policy e di un modulo da utilizzare per caricare un file allegato.

Costruzione di una policy e di un modulo

La policy seguente supporta caricamenti in Amazon S3 per il 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", ""]
  ]
}
```

Questa policy richiede quanto segue:

- Il caricamento deve avvenire prima delle 12:00, orario UTC, del 1 dicembre 2007.
- Il contenuto deve essere caricato nel bucket awsexamplebucket1.
- La chiave deve iniziare con "user/eric/".
- L'ACL è impostata su public-read.
- Il valore success_action_redirect è impostato su https://awsexamplebucket1.awsexamplebucket1.S3.us-west-1.amazonaws.com/successful_upload.html.
- L'oggetto è un file di immagine.
- Il x-amz-meta-uuid tag deve essere impostato su 14365123651274.
- Può x-amz-meta-tag contenere qualsiasi valore.

Di seguito è riportata una versione di questa policy con codifica Base64.

```
eyAiZXhwaXJhdGlvbiI6ICIyMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFoiLAogICJjb25kaXRpb25zIjogWwogICAgeyJidWN
```

Creare una firma utilizzando le credenziali personali. Ad esempio, `0RavWzkygo6QX9caELEqKi9kDbU=` è la firma per il documento di policy precedente.

Il modulo seguente supporta una richiesta POST per il bucket DOC-EXAMPLE-BUCKET che utilizza questa policy.

```
<html>
  <head>
    ...
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    ...
  </head>
  <body>
  ...
  <form action="https://DOC-EXAMPLE-BUCKET.s3.us-west-1.amazonaws.com/" method="post"
 enctype="multipart/form-data">
    Key to upload: <input type="input" name="key" value="user/eric/" /><br />
    <input type="hidden" name="acl" value="public-read" />
    <input type="hidden" name="success_action_redirect" value="https://
awsexamplebucket1.s3.us-west-1.amazonaws.com/successful_upload.html" />
    Content-Type: <input type="input" name="Content-Type" value="image/jpeg" /><br />
    <input type="hidden" name="x-amz-meta-uuid" value="14365123651274" />
    Tags for File: <input type="input" name="x-amz-meta-tag" value="" /><br />
    <input type="hidden" name="AWSAccessKeyId" value="AKIAIOSFODNN7EXAMPLE" />
    <input type="hidden" name="Policy" value="POLICY" />
    <input type="hidden" name="Signature" value="SIGNATURE" />
    File: <input type="file" name="file" /> <br />
    <!-- The elements after this will be ignored -->
    <input type="submit" name="submit" value="Upload to Amazon S3" />
  </form>
  ...
</html>
```

Richiesta di esempio

Questa richiesta presuppone che l'immagine caricata abbia una dimensione di 117.108 byte; i dati dell'immagine non sono inclusi.

```
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--
```

Risposta di esempio

```
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
```

Caricamento di un'area di testo

Argomenti

- Costruzione di una policy e di un modulo
- Richiesta di esempio
- Risposta di esempio

L'esempio seguente illustra il processo completo di costruzione di una policy e di un modulo per caricare un'area di testo. Il caricamento di un'area di testo è utile per inviare contenuti creati dall'utente, come i post dei blog.

Costruzione di una policy e di un modulo

La policy seguente supporta caricamenti di aree di testo in Amazon S3 per il 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", ""]
  ]
}
```

Questa policy richiede quanto segue:

- Il caricamento deve avvenire prima delle 12:00, orario GMT, del 01/12/2007.
- Il contenuto deve essere caricato nel bucket awsexamplebucket1.
- La chiave deve iniziare con "user/eric/".
- L'ACL è impostata su public-read.
- Il valore success_action_redirect è impostato su https://awsexamplebucket.awsexamplebuckett1.s3.us-west-1.amazonaws.com/new_post.html.
- L'oggetto è un testo HTML.
- Il x-amz-meta-uuid tag deve essere impostato su 14365123651274.
- Può x-amz-meta-tag contenere qualsiasi valore.

Di seguito è riportata una versione di questa policy con codifica Base64.

```
eyAiZXhwaXJhdGlvbiI6ICIyMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFoiLAogICJjb25kaXR
```

```
pb25zIjogWwogICAgeyJidWNrZXQiOiAiam9obnNtaXRoIn0sCiAgICBbInN0YXJ0cy13aXRoIiwgIiRrZXkiLCAidXNlci
LAogICAgeyJhY2wiOiAicHVibGljLXJlYWQifSwKICAgIHsic3VjY2Vzc19hY3Rpb25fcmVkaXJlY3QiOiAiaHR0cDovL2p
C5zMy5hbWF6b25hd3MuY29tL25ld19wb3N0Lmh0bWwifSwKICAgIFsiZXEiLCAiJENvbnRlbnQtVHlwZSIsICJ0ZXh0L2h0
CAgIHsieC1hbXotbWV0YS11dWlkIjogIjE0MzY1MTIzNjUxMjc0In0sCiAgICBbInN0YXJ0cy13aXRoIiwgIiR4LWFtei1t
IsICIiXQogIF0KfQo=
```

Creare una firma utilizzando le credenziali personali. Ad esempio, qA7FWXKq6VvU68lI9KdveT1cWgF= è la firma per il documento di policy precedente.

Il modulo seguente supporta una richiesta POST per il bucket DOC-EXAMPLE-BUCKET che utilizza questa policy.

```
<html>
  <head>
    ...
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    ...
  </head>
  <body>
  ...
  <form action="https://DOC-EXAMPLE-BUCKET.s3.us-west-1.amazonaws.com/" method="post"
 enctype="multipart/form-data">
    Key to upload: <input type="input" name="key" value="user/eric/" /><br />
    <input type="hidden" name="acl" value="public-read" />
    <input type="hidden" name="success_action_redirect" value="https://
awsexamplebucket1.s3.us-west-1.amazonaws.com/new_post.html" />
    <input type="hidden" name="Content-Type" value="text/html" />
    <input type="hidden" name="x-amz-meta-uuid" value="14365123651274" />
    Tags for File: <input type="input" name="x-amz-meta-tag" value="" /><br />
    <input type="hidden" name="AWSAccessKeyId" value="AKIAIOSFODNN7EXAMPLE" />
    <input type="hidden" name="Policy" value="POLICY" />
    <input type="hidden" name="Signature" value="SIGNATURE" />
    Entry: <textarea name="file" cols="60" rows="10">

Your blog post goes here.

    </textarea><br />
    <!-- The elements after this will be ignored -->
    <input type="submit" name="submit" value="Upload to Amazon S3" />
  </form>
  ...
</html>
```

Richiesta di esempio

Questa richiesta presuppone che l'immagine caricata abbia una dimensione di 117.108 byte; i dati dell'immagine non sono inclusi.

```
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--
```

Risposta di esempio

```
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

Questa sezione descrive come utilizzare POST con Adobe Flash.

Sicurezza di Adobe Flash Player

Per impostazione predefinita, il modello di sicurezza di Adobe Flash Player proibisce ad Adobe Flash Player di eseguire connessioni di rete con server al di fuori del dominio che serve il file SWF.

Per sostituire l'impostazione predefinita, è necessario caricare un file crossdomain.xml pubblicamente leggibile nel bucket che accetterà i caricamenti POST. Di seguito è riportato un esempio di file crossdomain.xml.

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
"http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
<allow-access-from domain="*" secure="false" />
</cross-domain-policy>
```

> **Note**
>
> Per ulteriori informazioni sul modello di sicurezza di Adobe Flash, visitare il sito Web di Adobe.
> L'aggiunta del file crossdomain.xml al bucket permette la connessione di qualsiasi strumento Adobe Flash Player al file crossdomain.xml all'interno del bucket, ma non concede l'accesso all'effettivo bucket Amazon S3.

Considerazioni su Adobe Flash

L' FileReference API di Adobe Flash aggiunge il campo del `Filename` modulo alla richiesta POST. Quando crei applicazioni Adobe Flash che vengono caricate su Amazon S3 utilizzando l'azione FileReference API, includi la seguente condizione nella tua policy:

```
['starts-with', '$Filename', '']
```

Alcune versioni di Adobe Flash Player non gestiscono correttamente le risposte HTTP che hanno un corpo vuoto. Per configurare POST in modo che restituisca una risposta senza corpo vuoto, impostare `success_action_status` su 201. Amazon S3 restituisce quindi un documento XML con codice di stato 201. Per informazioni sul contenuto del documento XML, consulta POST Object. Per informazioni sui campi del modulo, consulta Campi del modulo HTML.

Modelli di progettazione delle best practice: ottimizzazione delle prestazioni di Amazon S3

Le applicazioni possono facilmente raggiungere migliaia di transazioni al secondo come prestazioni delle richieste durante il caricamento e il recupero di risorse di storage da Amazon S3. Amazon S3 si ridimensiona automaticamente fino a tassi di richiesta elevati. Ad esempio, l'applicazione può ottenere almeno 3.500 richieste PUT/COPY/POST/DELETE o 5.500 richieste GET/HEAD al secondo per prefisso Amazon S3 partizionato. Non esistono limiti al numero di prefissi in un bucket. Puoi migliorare le prestazioni in lettura o scrittura utilizzando la parallelizzazione. Ad esempio, se si creano 10 prefissi in un bucket Amazon S3 per parallelizzare le letture, è possibile scalare le prestazioni di lettura a 55.000 richieste di lettura al secondo. Allo stesso modo, è possibile ridimensionare le operazioni di scrittura scrivendo su più prefissi. Il ridimensionamento, nel caso delle operazioni di lettura e scrittura, avviene gradualmente e non è istantaneo. Sebbene Amazon S3 stia eseguendo il dimensionamento alla nuova frequenza di richieste più elevata, si potrebbero verificare alcuni errori 503 (Slow Down). Questi errori scompariranno al termine del ridimensionamento. Per ulteriori informazioni sulla creazione e sull'utilizzo dei prefissi, consulta Organizzazione degli oggetti utilizzando i prefissi.

Alcune applicazioni di data lake in Amazon S3 analizzano milioni o miliardi di oggetti per query eseguite su diversi petabyte di dati. Queste applicazioni di data lake raggiungono tassi di trasferimento per singola istanza che massimizzano l'utilizzo dell'interfaccia di rete per l'istanza Amazon EC2, arrivando fino a 100 Gb/s su una singola istanza. Queste applicazioni poi aggregano throughput su più istanze per ottenere diversi terabit al secondo.

Altre applicazioni sono sensibili alla latenza, come le applicazioni di messaggistica sui social media. Queste applicazioni possono raggiungere latenze coerenti per piccoli oggetti (e first-byte-out latenze per oggetti più grandi) di circa 100-200 millisecondi.

Altri AWS servizi possono inoltre contribuire ad accelerare le prestazioni per diverse architetture applicative. Ad esempio, se desideri velocità di trasferimento più elevate su una singola connessione HTTP o latenze di un millisecondo, usa Amazon o Amazon per la memorizzazione nella cache ElastiCache con Amazon CloudFront S3.

Inoltre, se desideri trasferimenti dei dati veloci su lunghe distanze tra un client e un bucket S3, utilizza Configurazione di trasferimenti veloci e sicuri di file con Amazon S3 Transfer Acceleration. Transfer Acceleration utilizza le edge location distribuite a livello globale per accelerare il trasporto dei dati su distanze geografiche. CloudFront Se il tuo carico di lavoro Amazon S3 utilizza la crittografia lato

server con AWS KMS, consulta AWS KMS Limits nella AWS Key Management Service Developer Guide per informazioni sulle frequenze di richiesta supportate per il tuo caso d'uso.

Gli argomenti seguenti presentano le linee guida e i modelli di progettazione per le best practice per ottimizzare le prestazioni per le applicazioni che utilizzano Amazon S3. Per le informazioni più aggiornate sull'ottimizzazione delle prestazioni per Amazon S3, consulta Linee guida sulle prestazioni per Amazon S3 e Schemi di progettazione per le prestazioni di Amazon S3.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della classe di archiviazione Amazon S3 Express One Zone con bucket di directory, consulta Che cos'è S3 Express One Zone? e Bucks di directory.

Argomenti

Linee guida sulle prestazioni per Amazon S3

Per sviluppare applicazioni che caricano e recuperano oggetti da Amazon S3, segui le nostre linee guida sulle best practice per ottimizzare le prestazioni. Offriamo anche Schemi di progettazione delle prestazioni più dettagliati.

Per ottenere prestazioni ottimali per le applicazioni su Amazon S3, consigliamo di adottare le linee guida seguenti.

Argomenti

- Utilizzo della versione più recente degli SDK AWS

Misura le prestazioni

Quando ottimizzi le prestazioni, devi verificare i requisiti che riguardano il throughput della rete, la CPU e la DRAM. A seconda della combinazione di esigenze per queste risorse diverse, può essere utile valutare più tipi di istanza Amazon EC2. Per ulteriori informazioni sui tipi di istanza, consulta Instance Types nella Amazon EC2 User Guide.

Durante la misurazione delle prestazioni, è utile anche verificare i tempi, la latenza e la velocità del trasferimento dei dati DNS utilizzando strumenti di analisi HTTP.

Per comprendere i requisiti relativi alle prestazioni e ottimizzare le prestazioni dell'applicazione, puoi anche monitorare le risposte di errore 503 che ricevi. Il monitoraggio di determinate metriche delle prestazioni può comportare spese aggiuntive. Per ulteriori informazioni, consulta i Prezzi di Amazon S3.

Monitoraggio del numero di risposte all'errore di stato 503 (Rallentamento)

Per monitorare il numero di risposte all'errore di stato 503 che ricevi, puoi utilizzare una delle seguenti opzioni:

- Usa i parametri delle CloudWatch richieste Amazon per Amazon S3. Le metriche della CloudWatch richiesta includono una metrica per 5xx risposte allo stato. Per ulteriori informazioni sulle metriche delle CloudWatch richieste, consulta. Monitoraggio delle metriche con Amazon CloudWatch
- Utilizza il conteggio dell'errore 503 (Servizio non disponibile) disponibile nella sezione delle metriche avanzate di Amazon S3 Storage Lens. Per ulteriori informazioni, consulta Utilizzo dei parametri di S3 Storage Lens per migliorare le prestazioni.
- Utilizza la registrazione degli accessi al server Amazon S3 Con la registrazione degli accessi al server, puoi filtrare ed esaminare tutte le richieste che ricevono risposte 503 (Errore interno). Puoi anche utilizzare Amazon Athena per analizzare i log. Per ulteriori informazioni sulla registrazione degli accessi al server, consulta Registrazione delle richieste con registrazione dell'accesso al server.

Monitorando il numero del codice di errore di stato HTTP 503, spesso puoi ottenere informazioni dettagliate preziose su quali prefissi, chiavi o bucket ricevono il maggior numero di richieste di limitazione (della larghezza di banda della rete).

Scala le connessioni di storage orizzontalmente

Distribuire le richieste su più connessioni è uno schema di progettazione comune per scalare orizzontalmente le prestazioni. Se devi creare applicazioni ad alte prestazioni, pensa ad Amazon S3 come un sistema distribuito di dimensioni molto grandi, non un singolo endpoint di rete come un server di storage tradizionale. Puoi ottenere prestazioni ottimali inviando più richieste simultanee ad Amazon S3. Distribuisci queste richieste su connessioni separate per massimizzare la larghezza di banda accessibile da Amazon S3. Amazon S3 non impone limiti al numero di connessioni effettuate al bucket.

Utilizza i fetches Byte-Range

Utilizzando l'intestazione HTTP `Range` in una richiesta GET Object, puoi recuperare un intervallo di byte da un oggetto, trasferendo solo la parte specificata. Puoi utilizzare connessioni simultanee ad Amazon S3 per recuperare diversi intervalli di byte all'interno dello stesso oggetto. Questa operazione ti permette di ottenere un throughput aggregato superiore rispetto a una singola richiesta whole-object. Recuperare range minori da oggetti più grandi permette inoltre alla tua applicazione di migliorare i tempi di ripetizione quando le richieste sono interrotte. Per ulteriori informazioni, consulta Download di oggetti.

Le dimensioni tipiche per le richieste byte-range sono di 8 o 16 MB. Se gli oggetti sono oggetti PUT che utilizzano un caricamento in più parti, è buona pratica trasformarli in oggetti GET nelle stesse dimensioni della parte (o almeno allineati ai limiti della parte) per ottenere le prestazioni migliori. Le richieste GET possono rivolgersi direttamente alle singole parti; ad esempio, `GET ?partNumber=N.`

Riprova le richieste per le applicazioni sensibili alla latenza

I timeout e i tentativi aggressivi aiutano a mantenere la latenza uniforme. Poiché Amazon S3 opera su vasta scala, se la prima richiesta è lenta, un nuovo tentativo di richiesta adotterà un percorso diverso e riuscirà rapidamente. Gli AWS SDK dispongono di valori di timeout e riprova configurabili che puoi regolare in base alle tolleranze della tua applicazione specifica.

Combina Amazon S3 (storage) e Amazon EC2 (elaborazione) nello stesso Regione AWS

Sebbene i nomi dei bucket S3 siano univoci a livello globale, ogni bucket viene archiviato in una regione che sceglierai durante la creazione del bucket stesso. Per ottimizzare le prestazioni, ti

consigliamo di accedere al bucket dalle istanze Amazon EC2 nello stesso modo, quando possibile. Regione AWS Questa operazione permette di ridurre la latenza e i costi di trasferimento dei dati.

Per ulteriori informazioni sui costi dl trasferimento dei dati, consulta Prezzi di Amazon S3.

Utilizza Amazon S3 Transfer Acceleration per ridurre al minimo la latenza causata dalla distanza

Configurazione di trasferimenti veloci e sicuri di file con Amazon S3 Transfer Acceleration gestisce trasferimenti di file veloci, facili e sicuri su vaste distanze geografiche tra il client e un bucket S3. Transfer Acceleration sfrutta le edge location distribuite a livello globale di Amazon CloudFront. Quando arrivano in una edge location, i dati vengono instradati ad Amazon S3 attraverso un percorso di rete ottimizzato. Transfer Acceleration è ideale per il trasferimento regolare di gigabyte e terabyte di dati sui diversi continenti. È inoltre utile per i clienti che effettuano il caricamento in un bucket centralizzato da tutto il mondo.

Puoi utilizzare lo strumento Speed Comparison di Amazon S3 Transfer Acceleration per confrontare le velocità di caricamento accelerate e non accelerate tra regioni Amazon S3. Questo strumento utilizza caricamenti in più parti per trasferire un file dal browser in uso a diverse regioni Amazon S3 con e senza l'utilizzo di Amazon S3 Transfer Acceleration.

Utilizzo della versione più recente degli SDK AWS

Gli AWS SDK forniscono supporto integrato per molte delle linee guida consigliate per l'ottimizzazione delle prestazioni di Amazon S3. Gli SDK offrono un'API più semplice per utilizzare al meglio Amazon S3 internamente a un'applicazione e vengono aggiornati regolarmente in modo da seguire le best practice più recenti. Ad esempio, gli SDK includono la logica per riprovare automaticamente le richieste sugli errori HTTP 503 e investono in codice per rispondere e adattarsi a connessioni lente.

Gli SDK forniscono anche lo strumento TransferManager, che automatizza connessioni con dimensionamento orizzontale in modo da raggiungere migliaia di richieste al secondo, usando richieste di intervalli di byte laddove appropriato. È importante utilizzare la versione più recente degli AWS SDK per ottenere le più recenti funzionalità di ottimizzazione delle prestazioni.

Puoi inoltre ottimizzare le prestazioni quando utilizzi le richieste dell'API REST HTTP. Quando utilizzi l'API REST, devi seguire le stesse best practice che fanno parte degli SDK. Consenti i timeout e i tentativi sulle richieste lente e le connessioni multiple per permettere il recupero dei dati degli oggetti

in parallelo. Per informazioni sull'utilizzo dell'API REST, consulta la Documentazione di riferimento delle API di Amazon Simple Storage Service.

Schemi di progettazione per le prestazioni di Amazon S3

Nel progettare applicazioni per caricare e recuperare oggetti da Amazon S3, utilizza i nostri schemi di progettazione delle best practice per ottenere prestazioni ottimali per l'applicazione. Offriamo anche Linee guida sulle prestazioni, che puoi prendere in considerazione quando pianifichi l'architettura dell'applicazione.

Per ottimizzare le prestazioni, puoi utilizzare i seguenti schemi di progettazione.

Argomenti

- Utilizzo del caching per i contenuti ad accesso frequente
- Timeout e tentativi per applicazioni sensibili alla latenza
- Dimensionamento orizzontale e parallelizzazione delle richieste per throughput elevato
- Utilizzo di Amazon S3 Transfer Acceleration per accelerare trasferimenti di dati in zone geografiche lontane

Utilizzo del caching per i contenuti ad accesso frequente

Molte applicazioni che archiviano dati in Amazon S3 fungono da "working set" dei dati richiesti ripetutamente dagli utenti. Se un carico di lavoro invia richieste GET ripetute per un set comune di oggetti, puoi utilizzare una cache come Amazon CloudFront ElastiCache, Amazon o AWS Elemental MediaStoreper ottimizzare le prestazioni. L'adozione corretta di una cache può portare a una bassa latenza e a tassi di trasferimento dei dati più alti. Le applicazioni che utilizzano il caching inviano anche meno richieste dirette ad Amazon S3, riducendo i costi delle richieste.

Amazon CloudFront è una rete di distribuzione rapida dei contenuti (CDN) che memorizza in modo trasparente nella cache i dati di Amazon S3 in un ampio set di punti di presenza distribuiti geograficamente (). PoPs Quando è possibile accedere agli oggetti da più regioni o tramite Internet, CloudFront consente di memorizzare i dati nella cache in prossimità degli utenti che accedono agli oggetti. In questo modo, è possibile distribuire contenuti Amazon S3 comuni con prestazioni elevate. Per informazioni in merito CloudFront, consulta l'Amazon CloudFront Developer Guide.

Amazon ElastiCache è una cache gestita in memoria. Con ElastiCache, puoi effettuare il provisioning di istanze Amazon EC2 che memorizzano oggetti nella cache. Il caching porta alla riduzione di

grandezza della latenza GET e ad aumenti sostanziali nel throughput del download. Per utilizzarlo ElastiCache, è necessario modificare la logica dell'applicazione per popolare la cache con oggetti caldi e verificare la presenza di oggetti caldi nella cache prima di richiederli da Amazon S3. Per esempi di utilizzo ElastiCache per migliorare le prestazioni di Amazon S3 GET, consulta il post sul blog Turbocharge Amazon S3 with Amazon for Redis. ElastiCache

AWS Elemental MediaStore è un sistema di caching e distribuzione dei contenuti creato specificamente per i flussi di lavoro video e la distribuzione di contenuti multimediali da Amazon S3. MediaStore fornisce API end-to-end di archiviazione specifiche per i video ed è consigliato per carichi di lavoro video sensibili alle prestazioni. Per informazioni in merito MediaStore, consulta la Guida per l'utente.AWS Elemental MediaStore

Timeout e tentativi per applicazioni sensibili alla latenza

In alcune situazioni un'applicazione riceve una risposta da Amazon S3 che indica che è necessario un nuovo tentativo. Amazon S3 mappa i nomi dei bucket e degli oggetti ai dati degli oggetti a essi associati. Se un'applicazione genera alti tassi di richiesta (in genere tassi sostenuti di oltre 5.000 richieste al secondo per un piccolo numero di oggetti) potrebbe ricevere risposte di rallentamento HTTP 503. Se si verifica questo errore, ogni SDK AWS implementa la logica di tentativo automatica utilizzando il backoff esponenziale. Se non stai utilizzando un SDK AWS , quando ricevi un errore HTTP 503 è necessario implementare la logica di tentativo. Per informazioni sulle tecniche di back-off, vedere Error Retries and Exponential Backoff in. AWSRiferimenti generali di Amazon Web Services

Amazon S3 si ridimensiona automaticamente in risposta a nuovi tassi di richiesta prolungati, ottimizzando dinamicamente le prestazioni. Mentre Amazon S3 ottimizza internamente per sostenere un nuovo tasso di richieste, riceverai temporaneamente risposte di richiesta HTTP 503 fino al completamento dell'ottimizzazione. Dopo che Amazon S3 ha ottimizzato internamente le prestazioni per il nuovo tasso di richiesta, tutte le richieste vengono in genere gestite senza nuovi tentativi.

Per le applicazioni sensibili alla latenza, Amazon S3 consiglia di monitorare e ritentare in modo aggressivo le operazioni più lente. Nel ritentare una richiesta, consigliamo di utilizzare una nuova connessione ad Amazon S3 e di eseguire una nuova ricerca DNS.

Quando effettui richieste di dimensioni grandi e variabili (ad esempio, oltre 128 MB), consigliamo di tracciare il throughput raggiunto e di ritentare il 5 percento più lento delle richieste. Quando effettui richieste più piccole (ad esempio, meno di 512 KB) dove le latenze medie sono spesso dell'ordine di decine di millisecondi, una buona linea guida è ritentare un'operazione GET o PUT dopo 2 secondi. Se sono necessari tentativi aggiuntivi, la best practice è di effettuare il backoff. Ad

esempio, consigliamo di emettere un tentativo dopo 2 secondi e un secondo tentativo dopo 4 secondi aggiuntivi.

Se l'applicazione effettua richieste a dimensione fissa ad Amazon S3, il tempo di risposta per ogni richiesta sarà più costante. In questo caso, una strategia semplice è identificare l'1 percento più lento delle richieste e ritentarle. Anche un singolo tentativo è efficace nella riduzione della latenza.

Se utilizzi AWS Key Management Service (AWS KMS) per la crittografia lato server, consulta Limits nella AWS Key Management Service Developer Guide per informazioni sulle frequenze di richiesta supportate per il tuo caso d'uso.

Dimensionamento orizzontale e parallelizzazione delle richieste per throughput elevato

Amazon S3 è un sistema distribuito di dimensioni molto grandi. Per sfruttarne a pieno la capacità di dimensionamento, consigliamo di ridimensionare orizzontalmente le richieste parallele agli endpoint del servizio Amazon S3. Oltre a distribuire le richieste in Amazon S3, questo tipo di approccio al dimensionamento permette di distribuire il carico su più percorsi nella rete.

Per i trasferimenti a throughput elevato, Amazon S3 consiglia di utilizzare applicazioni che usano più connessioni a dati GET o PUT in parallelo. Ad esempio, questo è supportato da Amazon S3 Transfer Manager nell'SDK AWS Java e la maggior parte degli altri AWS SDK fornisce costrutti simili. Per alcune applicazioni, puoi raggiungere connessioni parallele lanciando simultaneamente richieste multiple in diversi thread dell'applicazione o in diverse istanze dell'applicazione. Il miglior approccio da adottare dipende dall'applicazione e dalla struttura degli oggetti a cui accedi.

Puoi utilizzare gli AWS SDK per emettere richieste GET e PUT direttamente anziché utilizzare la gestione dei trasferimenti nell'SDK. AWS Questo approccio ti permette di ottimizzare il carico di lavoro in modo più diretto senza rinunciare al supporto degli SDK per i tentativi e la gestione delle risposte HTTP 503 che potrebbero verificarsi. Come regola generale, quando scarichi oggetti di grandi dimensioni all'interno di una regione da Amazon S3 ad Amazon EC2, consigliamo di effettuare richieste simultanee di intervalli di byte di un oggetto in base a una granularità di 8–16 MB. Effettua una richiesta simultanea per ogni 85–90 MB/s di throughput di rete desiderato. Per saturare una network interface card (NIC) da 10 Gb/s, devi utilizzare circa 15 richieste simultanee su connessioni separate. Puoi scalare le richieste simultanee su più connessioni per saturare più rapidamente le NCI, come quelle da 25 Gb/s o 100 Gb/s.

La misurazione delle prestazioni è importante quando ottimizzi il numero di richieste da emettere simultaneamente. Consigliamo di iniziare con un richiesta alla volta. Misura la larghezza di banda

della rete raggiunta e l'uso delle altre risorse che la tua applicazione utilizza nell'elaborazione dei dati. Puoi quindi identificare la risorsa con un collo di bottiglia (ossia, la risorsa con l'utilizzo più elevato) e di conseguenza il numero di richieste che possono essere utili. Ad esempio, se elaborare una richiesta alla volta porta a un utilizzo della CPU del 25 percento, questo dato suggerisce che possono essere emesse fino a quattro richieste simultanee. La misurazione è essenziale ed è utile per confermare l'utilizzo della risorsa quando il tasso di richiesta aumenta.

Se l'applicazione invia richieste direttamente ad Amazon S3 utilizzando l'API REST, ti consigliamo di utilizzare un pool di connessioni HTTP e di riutilizzare ogni connessione per una serie di richieste. Evitare la configurazione della connessione per richiesta elimina la necessità di eseguire handshake slow-start su TCP e Secure Sockets Layer (SSL) su ogni richiesta. Per informazioni sull'utilizzo dell'API REST, consulta la Documentazione di riferimento delle API di Amazon Simple Storage Service.

Infine, è utile considerare con attenzione DNS e verificare accuratamente che le richieste vengano distribuite su un ampio pool di indirizzi IP di Amazon S3. Le query DNS per Amazon S3 passano per un elenco di grandi dimensioni di endpoint IP. Ma effettuare il caching dei resolver o del codice dell'applicazione che riutilizza un singolo indirizzo IP non trae vantaggio dalla diversità degli indirizzi e dal bilanciamento del carico che ne deriva. Utilità di rete come lo strumento a riga di comando `netstat` possono mostrare gli indirizzi IP utilizzati per la comunicazione con Amazon S3 e forniamo linee guida per le configurazioni DNS da utilizzare. Per ulteriori informazioni su queste linee guida, consulta Esecuzione di richieste.

Utilizzo di Amazon S3 Transfer Acceleration per accelerare trasferimenti di dati in zone geografiche lontane

Configurazione di trasferimenti veloci e sicuri di file con Amazon S3 Transfer Acceleration è efficace nel ridurre o eliminare la latenza causata dalla distanza geografica tra client lontani a livello globale e un'applicazione locale che utilizza Amazon S3. Transfer Acceleration utilizza le edge location distribuite a livello globale per il trasporto dei dati. CloudFront La rete AWS perimetrale ha punti di presenza in più di 50 località. Oggi viene utilizzato per distribuire contenuti CloudFront e fornire risposte rapide alle query DNS effettuate su Amazon Route 53.

La rete edge permette anche di accelerare i trasferimenti dei dati da e verso Amazon S3. È ideale per le applicazioni che trasferiscono i dati tra continenti, dispongono di connessioni a Internet veloci e utilizzano oggetti di grandi dimensioni o hanno molti contenuti da caricare. Quando arrivano in una edge location, i dati vengono instradati ad Amazon S3 su un percorso di rete ottimizzato. In generale,

più lontano ti trovi da una regione Amazon S3, maggiore è il miglioramento della velocità che otterrai utilizzando Transfer Acceleration.

Puoi configurare Transfer Acceleration su bucket nuovi o esistenti. Puoi utilizzare un endpoint Amazon S3 Transfer Acceleration separato per AWS utilizzare le edge location. Il modo migliore per verificare se Transfer Acceleration migliora le prestazioni delle richieste client consiste nell'utilizzare lo strumento Speed Comparison di Amazon S3 Transfer Acceleration. Le configurazioni e le condizioni della rete variano in base al momento e alla località. Vengono quindi addebitati solo i trasferimenti in cui Amazon S3 Transfer Acceleration può potenzialmente migliorare le prestazioni di caricamento. Per informazioni sull'utilizzo di Transfer Acceleration con diversi AWS SDK, consulta. Abilitazione e utilizzo di S3 Transfer Acceleration

Che cos'è Amazon S3 su Outposts?

AWS Outposts è un servizio completamente gestito che offre la stessa AWS infrastruttura, gli stessi AWS servizi, API e strumenti praticamente a qualsiasi data center, spazio di co-location o struttura locale per un'esperienza ibrida davvero coerente. AWS Outposts è ideale per carichi di lavoro che richiedono accesso a bassa latenza ai sistemi locali, elaborazione locale dei dati, residenza dei dati e migrazione di applicazioni con interdipendenze di sistema locali. Per ulteriori informazioni, consulta Che cos'è AWS Outposts? nella Guida per l'utente di AWS Outposts .

Con Amazon S3 su Outposts puoi creare bucket S3 in Outposts e archiviare e recuperare facilmente gli oggetti on-premise. S3 su Outposts fornisce una nuova classe di storage, `OUTPOSTS`, che utilizza le API Amazon S3 ed è progettata per archiviare i dati in modo durevole e ridondante su più dispositivi e server in Outposts. Comunichi con il bucket Outposts utilizzando un punto di accesso e una connessione di endpoint su un cloud privato virtuale (VPC).

Puoi utilizzare le stesse API e funzionalità sui bucket Outposts come per Amazon S3, comprese policy di accesso, crittografia e tagging. Puoi usare S3 su Outposts tramite AWS Management Console AWS Command Line Interface ,AWS CLI() AWS , SDK o API REST.

Come funziona S3 su Outposts

S3 su Outposts è un servizio di storage di oggetti che consente di archiviare dati come oggetti nei bucket in Outpost. Un oggetto è un file di dati e tutti i metadati che lo descrivono. Un bucket è un container per oggetti o file.

Per archiviare i dati in S3 su Outposts, per prima cosa devi creare un bucket. Quando crei il bucket, ne specifichi il nome e l'Outpost che contiene il bucket. Per accedere al bucket S3 su Outposts ed eseguire le operazioni sugli oggetti, è necessario creare e configurare un punto di accesso. È inoltre necessario creare un endpoint per instradare le richieste al punto di accesso.

Gli access point semplificano l'accesso ai dati per qualsiasi Servizio AWS applicazione del cliente che archivia dati in S3. I punti di accesso sono endpoint di rete denominati che vengono collegati a bucket che puoi usare per eseguire operazioni su oggetti, ad esempio `GetObject` e `PutObject`. Ogni punto di accesso dispone di autorizzazioni e controlli di rete distinti.

Puoi creare e gestire i tuoi bucket, access point ed endpoint S3 su Outposts utilizzando gli SDK o AWS Management Console l'API AWS CLI AWS REST. Per caricare e gestire oggetti nel tuo bucket S3 on Outposts, puoi utilizzare gli SDK o AWS CLI l'API AWS REST.

Regioni

Durante il AWS Outposts provisioning, tu o AWS crei una connessione service link che ricollega Outpost alla regione di origine prescelta o di Regione AWS Outposts per le operazioni sui bucket e la telemetria. Un Outpost si basa sulla connettività con la Regione AWS madre. Il rack Outposts non è progettato per operazioni o ambienti disconnessi con connettività limitata o assente. Per ulteriori informazioni, consulta Connettività dell'Outpost alle Regioni AWS nella Guida per l'utente di AWS Outposts .

Bucket

Un bucket è un container per gli oggetti archiviati in S3 su Outposts. Puoi archiviare un numero qualsiasi di oggetti in un bucket e avere fino a 100 bucket per account per Outpost.

Quando crei un bucket, inserisci un nome e scegli l'Outpost in cui risiede. Dopo avere creato un bucket, non è possibile modificare il nome del bucket o spostare il bucket in un Outpost diverso. I nomi dei bucket devono seguire le regole di denominazione dei bucket Amazon S3. In S3 on Outposts, i nomi dei bucket sono univoci per un Outpost e. Account AWS I bucket S3 su Outposts richiedono `outpost-id`, `account-id` e nome per identificarli.

Il seguente esempio mostra il formato Amazon Resource Name (ARN) per i bucket S3 su Outposts. L'ARN è composto dalla regione di residenza del tuo Outpost, dal tuo account Outpost, dall'ID Outpost e dal nome del bucket.

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/bucket/bucket-name
```

Ogni oggetto è contenuto in un bucket. È necessario utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outposts. Quando specifichi il bucket per le operazioni di oggetto, utilizza l'ARN o l'alias del punto di accesso. Per ulteriori informazioni sugli alias del punto di accesso, consulta Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3 su Outposts.

Nell'esempio seguente viene illustrato il formato ARN del punto di accesso per S3 su Outposts, che include `outpost-id`, `account-id` e il nome del punto di accesso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Per ulteriori informazioni sui bucket, consulta Utilizzo di bucket S3 su Outposts.

Oggetti

Gli oggetti sono le entità fondamentali archiviate in S3 su Outposts. sono composti da dati e metadata. I metadati sono invece un set di coppie nome-valore che descrivono l'oggetto. Queste coppie includono alcuni metadati di default, ad esempio la data dell'ultima modifica, e metadati HTTP standard, come `Content-Type`. È anche possibile specificare metadata personalizzati al momento dell'archiviazione dell'oggetto. Un oggetto viene identificato in modo univoco in un bucket tramite una chiave (nome).

Con Amazon S3 su Outposts, i dati degli oggetti vengono sempre archiviati nell'Outpost. Quando si AWS installa un rack Outpost, i dati rimangono locali rispetto a Outpost per soddisfare i requisiti di residenza dei dati. I tuoi oggetti non lasciano mai il tuo Outpost e non sono in una Regione AWS. Poiché AWS Management Console è ospitato in una regione, non puoi utilizzare la console per caricare o gestire oggetti in Outpost. Tuttavia, puoi utilizzare l'API REST, AWS Command Line Interface (AWS CLI) e gli AWS SDK per caricare e gestire gli oggetti tramite i tuoi punti di accesso.

Chiavi

Una chiave oggetto (o nome chiave) è l'identificatore univoco di un oggetto in un bucket. Per ogni oggetto in un bucket è presente esattamente una chiave. La combinazione di un bucket e una chiave identifica in modo univoco ciascun oggetto.

L'esempio seguente mostra il formato ARN per S3 sugli oggetti Outposts, che include il Regione AWS codice per la regione in cui è ospitato l'Outpost, l'ID, l' Account AWS ID Outpost, il nome del bucket e la chiave dell'oggetto:

```
arn:aws:s3-outposts:us-west-2:123456789012:outpost/ op-01ac5d28a6a232904/
bucket/example-s3-bucket1/object/myobject
```

Per ulteriori informazioni sulle chiavi degli oggetti, consulta Utilizzo di oggetti S3 su Outposts.

Funzione Controllo delle versioni S3

Puoi utilizzare il controllo delle versioni S3 nei bucket Outposts per conservare più versioni di un oggetto nello stesso bucket. Puoi utilizzare Controllo delle versioni S3 per conservare, recuperare e ripristinare qualsiasi versione di ogni oggetto archiviato nei bucket . Il controllo delle versioni S3 ti consente di eseguire il ripristino a seguito di errori dell'applicazione e operazioni non intenzionali degli utenti.

Per ulteriori informazioni, consulta Gestione del controllo delle versioni S3 per il bucket S3 su Outposts.

ID versione

Se abiliti il controllo delle versioni S3 in un bucket, S3 su Outposts genera un ID versione univoco per ciascun oggetto aggiunto al bucket. Gli oggetti già esistenti nel bucket al momento dell'attivazione del controllo delle versioni hanno un ID versione `null`. Se modificate questi (o altri) oggetti con altre operazioni, ad esempio, i nuovi oggetti ottengono un ID di PutObjectversione univoco.

Per ulteriori informazioni, consulta Gestione del controllo delle versioni S3 per il bucket S3 su Outposts.

Classe di storage e crittografia

S3 su Outposts offre una nuova classe di storage, S3 Outposts (`OUTPOSTS`). La classe di storage S3 Outposts è disponibile solo per gli oggetti archiviati in bucket su AWS Outposts. Se provi a utilizzare altre classi di storage S3 con S3 su Outposts, S3 su Outposts restituisce l'errore `InvalidStorageClass`.

Gli oggetti archiviati nella classe di storage S3 Outposts (`OUTPOSTS`) vengono crittografati utilizzando la crittografia lato server con chiavi di crittografia gestite di Amazon S3 (SSE-S3). Per ulteriori informazioni, consulta Crittografia dei dati in S3 su Outposts.

Policy del bucket

Una bucket policy è una policy basata sulle risorse AWS Identity and Access Management (IAM) che puoi utilizzare per concedere le autorizzazioni di accesso al tuo bucket e agli oggetti in esso contenuti. Solo il proprietario del bucket può associare una policy a un bucket. Le autorizzazioni allegate a un bucket si applicano a tutti gli oggetti del bucket di proprietà del proprietario del bucket. Le policy di bucket sono limitate a una dimensione di 20 KB.

Le policy di bucket utilizzano la sintassi delle policy IAM basata su JSON, che è lo standard di AWS. Puoi utilizzare policy di bucket per aggiungere o negare autorizzazioni per gli oggetti in un bucket. I criteri del bucket consentono o rifiutano le richieste in base agli elementi della policy. Questi elementi possono includere richiedente, operazioni S3, risorse e aspetti o condizioni della richiesta (ad esempio, l'indirizzo IP utilizzato per inviarla). Ad esempio, puoi creare una policy che conceda autorizzazioni per gli account per caricare oggetti in un bucket S3 su Outposts garantendo al contempo che il proprietario del bucket abbia il pieno controllo degli oggetti caricati. Per ulteriori informazioni, consulta Esempi di policy relative ai bucket di Amazon S3.

Nella policy di bucket puoi utilizzare caratteri jolly (`*`) negli ARN e altri valori per concedere autorizzazioni a un sottoinsieme di oggetti. Ad esempio, puoi controllare l'accesso a gruppi di oggetti che iniziano con un prefisso comune o terminano con una determinata estensione, come `.html`.

Punti di accesso S3 su Outposts

I punti di accesso S3 su Outposts sono endpoint di rete denominati con policy di accesso dedicate che descrivono come è possibile accedere ai dati utilizzando tale endpoint. I punti di accesso semplificano la gestione dell'accesso ai dati su vasta scala per set di dati condivisi in S3 su Outposts. I punti di accesso vengono collegati ai bucket che puoi usare per eseguire operazioni su oggetti S3, ad esempio `GetObject` e `PutObject`.

Quando specifichi il bucket per le operazioni di oggetto, utilizza l'ARN o l'alias del punto di accesso. Per ulteriori informazioni sugli alias del punto di accesso, consulta Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3 su Outposts.

Ogni punto di accesso dispone di autorizzazioni e controlli di rete distinti che S3 su Outposts applica per qualsiasi richiesta effettuata tramite il punto di accesso. Ogni access point applica una policy di access point personalizzata che funziona in combinazione con la policy di bucket collegata al bucket sottostante.

Per ulteriori informazioni, consulta Accesso a bucket e oggetti S3 su Outposts.

Caratteristiche di S3 su Outposts

Gestione degli accessi

Amazon S3 offre le caratteristiche per la verifica e la gestione dell'accesso ai bucket e agli oggetti. Per impostazione predefinita, i bucket S3 su Outposts e gli oggetti al loro interno sono privati. Puoi accedere solo alle risorse S3 su Outposts che hai creato.

Per concedere autorizzazioni granulari per le risorse che supportano il tuo caso d'uso specifico o per verificare le autorizzazioni delle tue risorse Amazon S3, puoi utilizzare le seguenti caratteristiche.

- Blocco dell'accesso pubblico di S3: blocca l'accesso pubblico a bucket e oggetti. Per i bucket su Outposts, il blocco dell'accesso pubblico è sempre abilitato per impostazione predefinita.
- AWS Identity and Access Management (IAM): IAM è un servizio web che ti aiuta a controllare in modo sicuro l'accesso alle AWS risorse, incluse le risorse S3 on Outposts. Con IAM, puoi gestire a livello centrale le autorizzazioni che controllano le risorse AWS a cui possono accedere gli utenti. Utilizza IAM per controllare chi è autenticato (accesso effettuato) e autorizzato (dispone di autorizzazioni) per l'utilizzo di risorse.
- Punti di accesso S3 su Outposts: gestisci l'accesso ai dati per set di dati condivisi in S3 su Outposts. I punti di accesso sono denominati endpoint di rete con policy di accesso dedicate. I punti di accesso vengono collegati ai bucket che puoi usare per eseguire operazioni su oggetti, ad esempio `GetObject` e `PutObject`.
- Policy di bucket: utilizza il linguaggio delle policy basato su IAM per configurare le autorizzazioni basate sulle risorse per i bucket S3 e gli oggetti in essi contenuti.
- AWS Resource Access Manager (AWS RAM) — Condividi in modo sicuro la capacità di S3 on Outposts all'interno dell'organizzazione o delle unità organizzative (OU) all'interno dell'organizzazione o delle unità organizzative (OU) in Account AWS. AWS Organizations

Registrazione e monitoraggio dell'archiviazione

S3 su Outposts fornisce strumenti di registrazione e monitoraggio che puoi utilizzare per monitorare e controllare come vengono utilizzate le tue risorse S3 su Outposts. Per ulteriori informazioni, Strumenti di monitoraggio.

- CloudWatch Parametri Amazon per S3 on Outposts: monitora lo stato operativo delle tue risorse e comprendi la disponibilità della capacità.
- CloudWatch Eventi Amazon Events per S3 su Outposts: crea una regola per qualsiasi evento API S3 on Outposts per ricevere notifiche tramite CloudWatch tutti i target Events supportati, tra cui Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS) e. AWS Lambda
- AWS CloudTrail log per S3 su Outposts: registra le azioni intraprese da un utente, da un ruolo o da un utente in Servizio AWS S3 su Outposts. CloudTrail i log forniscono un tracciamento dettagliato delle API per le operazioni S3 a livello di bucket e a livello di oggetto.

Forte coerenza

In generale, S3 on Outposts offre una read-after-write forte coerenza per le richieste PUT e DELETE degli oggetti nel bucket S3 on Outposts. Regioni AWS Questo comportamento vale sia per le scritture dei nuovi oggetti che per le richieste PUT che sovrascrivono gli oggetti esistenti e le richieste DELETE. Inoltre, i tag dell'oggetto S3 su Outposts e i metadati dell'oggetto (ad esempio, l'oggetto HEAD) sono fortemente coerenti. Per ulteriori informazioni, consulta Modello di consistenza dati Amazon S3.

Servizi correlati

Una volta caricati i dati in S3 su Outposts, è possibile utilizzarli con altri Servizi AWS. Di seguito vengono riportati i servizi che potresti utilizzare più di frequente:

- Amazon Elastic Compute Cloud (Amazon EC2): fornisce capacità di calcolo scalabile e sicura in Cloud AWS. L'utilizzo Amazon EC2 riduce la necessità di investimenti anticipati in hardware e ti permette di sviluppare e distribuire più rapidamente le applicazioni. Puoi utilizzare Amazon EC2 per avviare il numero di server virtuali necessari, configurare la sicurezza e i servizi di rete, nonché gestire l'archiviazione.
- Amazon Elastic Block Store (Amazon EBS) su Outposts: utilizza snapshot locali Amazon EBS su Outposts per archiviare snapshot di volumi in un Outpost localmente in S3 su Outposts.
- Amazon Relational Database Service (Amazon RDS) su Outposts: utilizza i backup locali Amazon RDS per archiviare i backup Amazon RDS localmente nel tuo Outpost.
- AWS DataSync— Automatizza il trasferimento dei dati tra i tuoi Outposts e Regioni AWS scegli cosa trasferire, quando trasferire e quanta larghezza di banda di rete utilizzare. S3 on Outposts è integrato con. AWS DataSync Per le applicazioni locali che richiedono un'elaborazione locale ad alta velocità effettiva, S3 su Outposts fornisce archiviazione locale di oggetti in modo da ridurre al minimo i trasferimenti di dati e il buffer dalle variazioni di rete, offrendo al contempo la possibilità di trasferire facilmente i dati tra Outposts e le Regioni AWS.

Accesso a S3 su Outposts

Puoi lavorare con S3 su Outposts nei modi descritti di seguito:

AWS Management Console

La console è un'interfaccia utente basata sul Web per la gestione delle risorse S3 su Outposts e AWS . Se ti sei registrato a un account Account AWS, puoi accedere a S3 su Outposts accedendo e scegliendo S3 dalla AWS Management Console home page. AWS Management Console Quindi, scegli Outposts buckets (Bucket Outposts) dal riquadro di navigazione a sinistra.

AWS Command Line Interface

Puoi usare gli strumenti della AWS riga di comando per impartire comandi o creare script dalla riga di comando del tuo sistema per eseguire attività AWS (incluso S3).

Il AWS Command Line Interface (AWS CLI) fornisce comandi per un ampio set di. Servizi AWS AWS CLI È supportato su Windows, macOS e Linux. Per iniziare, consulta la AWS Command Line Interface Guida per l'utente di . Per ulteriori informazioni sui comandi utilizzabili con S3 su Outposts, consultas3api, s3control e s3outposts nella Documentazione di riferimento per i comandi AWS CLI .

AWS SDK

AWS fornisce SDK (kit di sviluppo software) costituiti da librerie e codice di esempio per vari linguaggi e piattaforme di programmazione (Java, Python, Ruby, .NET, iOS, Android e così via). Gli AWS SDK offrono un modo pratico per creare un accesso programmatico a S3 su Outposts e. AWS Dal momento che S3 on Outposts utilizza gli stessi SDK di Amazon S3, offre un'esperienza coerente utilizzando API, automazione e strumenti di S3.

S3 on Outposts è un servizio REST. Puoi inviare le richieste a S3 su Outposts utilizzando le librerie di SDK AWS che eseguono il wrapping dell'API REST sottostante, semplificando le attività di programmazione. Ad esempio, gli SDK si occupano di attività quali il calcolo delle firme, la firma crittografica delle richieste, la gestione degli errori e la ripetizione automatica delle richieste. Per informazioni sugli AWS SDK, incluso come scaricarli e installarli, consulta Tools to Build on. AWS

Pagamento per S3 su Outposts

Puoi acquistare una varietà di configurazioni AWS Outposts rack con una combinazione di tipi di istanze Amazon EC2, volumi SSD (Solid State Drive) di Amazon EBS General Purpose gp2 () e S3 on Outposts. I prezzi includono consegna, installazione e manutenzione del servizio dell'infrastruttura , patch e aggiornamenti software .

Per ulteriori informazioni, consulta Prezzi del rack AWS Outposts.

Passaggi successivi

Per ulteriori informazioni sull'utilizzo di S3 su Outposts, consulta i seguenti argomenti:

- Configurazione di Outpost
- In che modo Amazon S3 su Outposts è diverso da Amazon S3?
- Nozioni di base su Amazon S3 su Outposts
- Reti per S3 su Outposts
- Utilizzo di bucket S3 su Outposts
- Utilizzo di oggetti S3 su Outposts
- Sicurezza in S3 su Outposts
- Gestione dello storage S3 su Outposts
- Sviluppo con Amazon S3 su Outposts

Configurazione di Outpost

Per iniziare a utilizzare Amazon S3 su Outposts, hai bisogno di un Outpost con capacità Amazon S3 distribuita presso la tua struttura. Per informazioni sulle opzioni per ordinare una capacità outpost e S3, vedere AWS Outposts. Per verificare se Outposts ha capacità S3, puoi utilizzare la chiamata API ListOutpostsWithS3. Per le specifiche e per vedere in che modo S3 su Outposts è diverso da Amazon S3, consulta In che modo Amazon S3 su Outposts è diverso da Amazon S3?.

Per ulteriori informazioni, consulta i seguenti argomenti.

Argomenti

- Ordine di un nuovo Outpost

Ordine di un nuovo Outpost

Se hai bisogno di ordinare un nuovo Outpost con capacità S3, consulta Prezzi del rack AWS Outposts per comprendere l'opzione di capacità per Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Block Store (Amazon EBS) e Amazon S3.

Dopo aver selezionato la configurazione, attenersi alla procedura descritta in Creare un outpost e ordinare la capacità dell'outpost nella Guida dell'utente AWS Outposts.

In che modo Amazon S3 su Outposts è diverso da Amazon S3?

Amazon S3 su Outposts fornisce l'archiviazione di oggetti nell'ambiente AWS Outposts on-premise. S3 su Outposts ti consente di soddisfare le esigenze di elaborazione locale, residenza dei dati e prestazioni elevate mantenendo i dati vicini alle applicazioni on-premise. Utilizzando le API e le caratteristiche di Amazon S3, S3 su Outposts semplifica l'archiviazione, la protezione, l'assegnazione di tag, la creazione di report e il controllo dell'accesso ai dati su Outposts ed estende l'infrastruttura AWS alla tua struttura on-premise per un'esperienza ibrida coerente.

Per ulteriori informazioni sull'utilizzo di S3 su Outposts, consulta i seguenti argomenti.

Argomenti

Specifiche di S3 su Outposts

- La dimensione massima dei bucket Outposts è 50 TB.
- Il numero massimo di bucket Outposts per Account AWS è 100.
- I bucket Outposts sono accessibili solo tramite punti di accesso ed endpoint.
- Il numero massimo di access point per ogni bucket Outposts è 10.
- Le policy access point sono limitate a una dimensione di 20 KB.
- Il proprietario dell'Outpost può gestire l'accesso all'interno dell'organizzazione in AWS Organizations tramite AWS Resource Access Manager. Tutti gli account che devono accedere all'Outpost devono essere all'interno della stessa organizzazione dell'account proprietario in AWS Organizations.
- L'account proprietario del bucket S3 su Outposts è sempre il proprietario di tutti gli oggetti nel bucket.
- Solo l'account proprietario del bucket S3 su Outposts può eseguire operazioni sul bucket.

- Le limitazioni relative alle dimensioni degli oggetti sono coerenti con Amazon S3.
- Tutti gli oggetti archiviati in S3 su Outposts vengono archiviati nella classe di storage `OUTPOSTS`.
- Per impostazione predefinita, tutti gli oggetti archiviati nella classe di archiviazione `OUTPOSTS` vengono memorizzati utilizzando la crittografia lato server con chiavi di crittografia gestite da Amazon S3 (SSE-S3). Puoi inoltre scegliere di archiviare esplicitamente gli oggetti utilizzando la crittografia lato server con chiavi di crittografia fornite dal cliente (SSE-C).
- Se non c'è spazio sufficiente per archiviare un oggetto sul tuo Outpost, l'API restituirà un'eccezione di capacità insufficiente (ICE).

Operazioni API supportate da S3 su Outposts

Per un elenco di operazioni API supportate da S3 su Outposts, consulta Operazioni API in Amazon S3 su Outposts.

Funzionalità Amazon S3 non supportate da S3 su Outposts

Le seguenti funzionalità Amazon S3 al momento non sono supportate da Amazon S3 su Outposts. Tutti i tentativi di utilizzarle vengono rifiutati.

- Liste di controllo accessi di rete (ACL)
- Cross-Origin Resource Sharing (CORS)
- Operazioni in batch S3
- Report di inventario Amazon S3
- Modifica della crittografia predefinita del bucket
- Bucket pubblici
- Eliminazione dell'autenticazione a più fattori (MFA)
- Transizioni del ciclo di vita di Amazon S3 (a parte l'eliminazione degli oggetti e l'arresto dei caricamenti in più parti incompleti)
- Blocco di carattere legale del blocco oggetti S3
- Conservazione Blocco oggetto
- Crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS)
- S3 Replication Time Control (S3 RTC)
- Parametri delle richieste Amazon CloudWatch

- Configurazione dei parametri
- Transfer Acceleration
- Notifiche di eventi di Amazon S3
- Bucket con Pagamento a carico del richiedente
- S3 Select
- Eventi AWS Lambda
- Server access logging (Registrazione degli accessi al server)
- Richieste POST HTTP
- SOAP
- Accesso al sito web

Requisiti di rete di S3 su Outposts

- Per instradare le richieste a un punto di accesso S3 su Outposts, è necessario creare e configurare un endpoint S3 su Outposts. I seguenti limiti si applicano agli endpoint per S3 su Outposts:
  - Ogni cloud privato virtuale (VPC) su un Outpost può avere un endpoint associato, per un massimo di 100 endpoint per Outpost.
  - Puoi mappare più punti di accesso sullo stesso endpoint.
  - Gli endpoint possono essere aggiunti ai VPC solo con blocchi CIDR nei sottospazi dei seguenti intervalli CIDR:
    - 10.0.0.0/8
    - 172.16.0.0/12
    - 192.168.0.0/16
- Gli endpoint di un Outpost possono essere creati solo da VPC che presentano blocchi CIDR che non si sovrappongono.
- Un endpoint può essere creato solo dall'interno della propria sottorete Outposts.
- La sottorete che utilizzi per creare un endpoint deve contenere quattro indirizzi IP utilizzabili da Amazon S3 su Outposts.
- Il pool di indirizzi IP di proprietà del cliente (pool CoIP), se specificato, deve contenere quattro indirizzi IP utilizzabili da Amazon S3 su Outposts.
- È possibile creare un solo endpoint per Outpost per VPC.

Nozioni di base su Amazon S3 su Outposts

Con Amazon S3 su Outposts è possibile creare bucket S3 su AWS Outposts, nonché archiviare e recuperare facilmente gli oggetti on-Premise per le applicazioni che richiedono l'accesso ai dati in locale, l'elaborazione dei dati in locale e la residenza dei dati. S3 su Outposts fornisce una nuova classe di storage, S3 Outposts (OUTPOSTS), che utilizza le API Amazon S3 ed è progettata per archiviare i dati in modo durevole e ridondante su più dispositivi e server su AWS Outposts. Comunichi con il bucket Outpost utilizzando un punto di accesso e una connessione di endpoint su un Virtual Private Cloud (VPC). Sui bucket Outposts puoi utilizzare le stesse API e caratteristiche di Amazon S3, comprese policy di accesso, crittografia e tagging. Puoi utilizzare S3 su Outposts tramite la AWS Management Console, AWS Command Line Interface (AWS CLI), gli SDK AWS o l'API REST.

Con Amazon S3 su Outposts è possibile utilizzare le API e le funzionalità Amazon S3, ad esempio l'archiviazione degli oggetti, le policy di accesso, la crittografia e l'aggiunta di tag, su AWS Outposts come si fa su Amazon S3. Per informazioni su S3 su Outposts, consulta Che cos'è Amazon S3 su Outposts?.

Argomenti

- Configurazione di IAM con S3 su Outposts
- Nozioni di base per l'utilizzo di AWS Management Console
- Guida introduttiva all'utilizzo di AWS CLI and SDK for Java

Configurazione di IAM con S3 su Outposts

AWS Identity and Access Management (IAM) è un software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM controllano chi può autenticarsi (eseguire l'accesso) ed è autorizzato (dispone di autorizzazioni) a utilizzare le risorse di Amazon S3 su Outpost. IAM è un Servizio AWS il cui uso non comporta costi aggiuntivi. Per impostazione predefinita, gli utenti non dispongono delle autorizzazioni per le risorse e le operazioni di S3 su Outposts. Per concedere le autorizzazioni di accesso per S3 sulle risorse e le operazioni API di Outposts, puoi utilizzare IAM per creare utenti, gruppi o ruoli a cui associare le autorizzazioni.

Per fornire l'accesso, aggiungi autorizzazioni ai tuoi utenti, gruppi o ruoli:

- Utenti e gruppi in AWS IAM Identity Center:

Crea un set di autorizzazioni. Segui le istruzioni riportate nella pagina Create a permission set (Creazione di un set di autorizzazioni) nella Guida per l'utente di AWS IAM Identity Center .

- Utenti gestiti in IAM tramite un provider di identità:

  Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina Creating a role for a third-party identity provider (federation) (Creazione di un ruolo per un provider di identità di terze parti [federazione]) nella Guida per l'utente di IAM.

- Utenti IAM:
  - Crea un ruolo che l'utente possa assumere. Per istruzioni, consulta la pagina Creating a role for an IAM user (Creazione di un ruolo per un utente IAM) nella Guida per l'utente di IAM.
  - (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate nella pagina Aggiunta di autorizzazioni a un utente (console) nella Guida per l'utente di IAM.

Oltre alle policy IAM basate sull'identità, S3 su Outposts supporta sia le policy del bucket che le policy dei punti di accesso. Le policy del bucket e le policy del punto di accesso sono policy basate sulle risorsecollegate alla risorsa S3 su Outposts.

- Una policy del bucket è collegata al bucket e consente o nega le richieste al bucket e agli oggetti in esso contenuti in base agli elementi nella policy.
- Al contrario, una policy del punto di accesso è collegata al punto di accesso e consente o nega le richieste al punto di accesso.

La policy del punto di accesso funziona con la policy del bucket collegata al bucket S3 su Outposts sottostante. Affinché un'applicazione o un utente possa accedere agli oggetti in un bucket S3 su Outposts tramite un punto di accesso S3 su Outposts, sia la policy del punto di accesso che la policy del bucket devono consentire la richiesta.

Le limitazioni incluse in una policy di access point si applicano solo alle richieste effettuate tramite quell'access point. Ad esempio, se un punto di accesso è collegato a un bucket, non potrai utilizzare la policy del punto di accesso per consentire o negare le richieste che vengono effettuate direttamente al bucket. Tuttavia, le restrizioni applicate a una policy del bucket possono consentire o rifiutare le richieste effettuate direttamente al bucket o tramite il punto di accesso.

In una policy IAM o in una policy basata su risorse, definisci quali operazioni S3 su Outposts saranno consentite o negate. Le operazioni S3 su Outposts corrispondono a operazioni API S3 su Outposts

specifiche. Le operazioni S3 su Outposts utilizzano il prefisso dello spazio dei nomi `s3-outposts:`. Le richieste effettuate all'API di controllo S3 on Outposts in Regione AWS un e le richieste effettuate agli endpoint dell'API oggetto su Outpost vengono autenticate utilizzando IAM e autorizzate tramite il prefisso namespace. `s3-outposts:` Configurare gli utenti IAM e autorizzarli a fronte dello spazio dei nomi IAM `s3-outposts:` per lavorare con S3 su Outposts.

Per ulteriori informazioni, consulta Operazioni, risorse e chiavi di condizione per Amazon S3 su Outposts nella Documentazione di riferimento per l'autorizzazione ai servizi.

> **Note**
>
> - Le liste di controllo degli accessi (ACL) non sono supportate in S3 su Outposts.
> - Per impostazione predefinita, S3 su Outposts definisce il proprietario del bucket come proprietario dell'oggetto per avere la certezza che al proprietario di un bucket non possa essere impedito di accedere o eliminare oggetti.
> - S3 su Outposts dispone sempre di accesso pubblico blocco S3 abilitato per garantire che gli oggetti non possano mai avere accesso pubblico.

Per ulteriori informazioni sulla configurazione di IAM per S3 su Outposts, consulta i seguenti argomenti.

Argomenti

- Principi per le policy di S3 su Outposts
- ARN delle risorse per S3 su Outposts
- Esempi di policy per S3 su Outposts
- Autorizzazioni per endpoint S3 su Outposts
- Ruoli collegati ai servizi per S3 su Outposts

Principi per le policy di S3 su Outposts

Quando crei una policy basata su risorse per concedere l'accesso al bucket S3 su Outposts, devi utilizzare l'elemento `Principal` per specificare la persona o l'applicazione che può effettuare una richiesta per un'azione o un'operazione su tale risorsa. Per le policy S3 su Outposts, puoi utilizzare uno dei seguenti principali:

- Un Account AWS
- Un utente IAM
- Un ruolo IAM:
- Tutti i principali, specificando un carattere jolly (*) in una policy che utilizza un elemento `Condition` per limitare l'accesso a un intervallo IP specifico

> **Important**
>
> Non puoi scrivere una policy per un bucket S3 su Outposts che utilizza un carattere jolly (*) nell'elemento `Principal` a meno che la policy non includa anche una `Condition` che limita l'accesso a un intervallo di indirizzi IP specifico. Questa limitazione garantisce che non vi sia alcun accesso pubblico al bucket S3 su Outposts. Per vedere un esempio, consulta Esempi di policy per S3 su Outposts.

Per ulteriori informazioni sull'elemento `Principal`, consulta Elementi della policy JSON di AWS : principale nella Guida per l'utente di IAM.

ARN delle risorse per S3 su Outposts

Amazon Resource Names (ARN) per S3 on Outposts contengono l'ID Outpost oltre a Regione AWS quello su cui è ospitato l'Outpost, l'ID e il Account AWS nome della risorsa. Per accedere ed eseguire operazioni sui bucket e sugli oggetti Outposts, è necessario utilizzare uno dei formati ARN mostrati nella tabella seguente.

Il *`partition`* valore nell'ARN si riferisce a un gruppo di. Regioni AWS Ciascuno Account AWS è limitato a una partizione. Di seguito sono riportate le partizioni supportate:

- `aws` – Regioni AWS
- `aws-us-gov`— Regioni AWS GovCloud (US)

Formati ARN di S3 su Outposts

Amazon S3 su ARN Outposts.	Formato ARN	Esempio
Bucket ARN	`arn:`*`partition`*`:s3-outposts:`*`region`*`:`	`arn:`*`aws`*`:s3-outposts:`*`us-west-2`*

Amazon S3 su ARN Outposts.	Formato ARN	Esempio
	arn:*partition* :s3-outposts: *region*: *account_id* :outpost/ *outpost_id* /bucket/*bucket_name*	arn:*aws*:s3-outposts: *us-west-2* :*123456789012* :outpost/ *op-01ac5d28a6a232904* /bucket/*example-s3-bucket1*
ARN del punto di accesso	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*
Oggetto ARN	arn:*partition* :s3-outposts: *region*: *account_id* :outpost/ *outpost_id* /bucket/*bucket_name* /object/*object_key*	arn:*aws*:s3-outposts: *us-west-2* :*123456789012* :outpost/ *op-01ac5d28a6a232904* /bucket/*example-s3-bucket1 /object/myobject*
ARN dell'oggetto punto di accesso S3 su Outpost (utilizzato nelle policy)	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*

Amazon S3 su ARN Outposts.	Formato ARN	Esempio
ARN di S3 su Outposts	arn:*partition* :s3-outposts: *region*: *account_id* :outpost/ *outpost_id*	arn:*aws*:s3-outposts: *us-west-2* :*123456789012* :outpost/ *op-01ac5d28a6a232904*

Esempi di policy per S3 su Outposts

Example : policy sui bucket di S3 on Outposts con un preside Account AWS

La seguente policy sui bucket utilizza un Account AWS principale per concedere l'accesso a un bucket S3 on Outposts. Per utilizzare questa policy del bucket, sostituisci *user input placeholders* con le tue informazioni.

```
{
   "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 : policy del bucket S3 su Outposts policy con principale e chiave di condizione con carattere jolly (*****) per limitare l'accesso a un intervallo di indirizzi IP specifico.

La seguente policy del bucket utilizza un principale con carattere jolly (*) con la condizione `aws:SourceIp` per limitare l'accesso a un intervallo di indirizzi IP specifico. Per utilizzare questa policy del bucket, sostituisci *user input placeholders* con le tue informazioni.

```
{
    "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"
                }
            }
        }
    ]
}
```

Autorizzazioni per endpoint S3 su Outposts

S3 su Outposts richiede proprie autorizzazioni in IAM per gestire le operazioni degli endpoint S3 su Outposts.

> **Note**
>
> - Per gli endpoint che utilizzano il tipo di accesso del pool di indirizzi IP (pool CoIP) di proprietà del cliente, è inoltre necessario disporre delle autorizzazioni per lavorare con gli indirizzi IP del pool CoIP, come descritto nella tabella seguente.
> - Per gli account condivisi che accedono a S3 su Outposts AWS Resource Access Manager utilizzando, gli utenti di questi account condivisi non possono creare i propri endpoint su una sottorete condivisa. Se un utente in un account condiviso desidera gestire i propri endpoint, l'account condiviso deve creare una propria sottorete nell'Outpost. Per ulteriori informazioni, consulta the section called “Condivisione di S3 su Outposts”.

Autorizzazioni IAM correlate agli endpoint S3 su Outposts

Operazione	Autorizzazioni IAM
`CreateEndpoint`	`s3-outposts:CreateEndpoint` `ec2:CreateNetworkInterface` `ec2:DescribeNetworkInterfaces` `ec2:DescribeVpcs` `ec2:DescribeSecurityGroups` `ec2:DescribeSubnets` `ec2:CreateTags` `iam:CreateServiceLinkedRole` Per gli endpoint che utilizzano il tipo di accesso del pool di indirizzi IP di proprietà del cliente on-premise (pool CoIP), sono necessarie le seguenti autorizzazioni aggiuntive: `s3-outposts:CreateEndpoint` `ec2:DescribeCoipPools` `ec2:GetCoipPoolUsage` `ec2:AllocateAddress` `ec2:AssociateAddress` `ec2:DescribeAddresses` `ec2:DescribeLocalGatewayRou teTableVpcAssociations`
`DeleteEndpoint`	`s3-outposts:DeleteEndpoint` `ec2:DeleteNetworkInterface`

Operazione	Autorizzazioni IAM
	`ec2:DescribeNetworkInterfaces` Per gli endpoint che utilizzano il tipo di accesso del pool di indirizzi IP di proprietà del cliente on-premise (pool CoIP), sono necessarie le seguenti autorizzazioni aggiuntive: `s3-outposts:DeleteEndpoint` `ec2:DisassociateAddress` `ec2:DescribeAddresses` `ec2:ReleaseAddress`
`ListEndpoints`	`s3-outposts:ListEndpoints`

> **Note**
>
> Puoi utilizzare i tag delle risorse in una policy IAM per gestire le autorizzazioni.

Ruoli collegati ai servizi per S3 su Outposts

S3 su Outposts utilizza ruoli IAM collegati ai servizi per creare alcune risorse di rete per tuo conto. Per ulteriori informazioni, consulta Utilizzo dei ruoli collegati ai servizi per Amazon S3 su Outposts.

Nozioni di base per l'utilizzo di AWS Management Console

Con Amazon S3 su Outposts è possibile creare bucket S3 su AWS Outposts, nonché archiviare e recuperare facilmente gli oggetti on-Premise per le applicazioni che richiedono l'accesso ai dati in locale, l'elaborazione dei dati in locale e la residenza dei dati. S3 su Outposts fornisce una nuova classe di archiviazione, S3 Outposts (`OUTPOSTS`), che utilizza le API Amazon S3 ed è progettata per archiviare i dati in modo durevole e ridondante su più dispositivi e server su AWS Outposts. Comunichi con il bucket Outpost utilizzando un punto di accesso e una connessione di endpoint su un Virtual Private Cloud (VPC). Sui bucket Outposts puoi utilizzare le stesse API e caratteristiche di Amazon S3, comprese policy di accesso, crittografia e tagging. Puoi utilizzare S3 su Outposts

tramite la AWS Management Console, AWS Command Line Interface (AWS CLI), gli SDK AWS o l'API REST. Per ulteriori informazioni, consulta Che cos'è Amazon S3 su Outposts?

Per iniziare a utilizzare S3 su Outposts con la console, consulta i seguenti argomenti. Per le nozioni di base su AWS CLI o AWS SDK for Java, consulta Guida introduttiva all'utilizzo di AWS CLI and SDK for Java.

Argomenti

Creazione di un bucket, un punto di accesso e un endpoint

La procedura seguente mostra come creare il primo bucket in S3 su Outposts. La prima volta che crei un bucket utilizzando la console, crei anche un punto di accesso e un endpoint associato al bucket in modo da poter iniziare immediatamente ad archiviare gli oggetti nel bucket.

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Seleziona Crea bucket Outposts.
4. In Bucket name (Nome bucket), immettere un nome conforme a DNS (Domain Name System) per il bucket.

   Il nome del bucket deve:

   - Essere univoco nell'Account AWS, nell'Outpost e nella Regione AWS in cui si trova l'Outpost.
   - Contenere da 3 a 63 caratteri.
   - Non contiene caratteri maiuscoli.
   - Iniziare con una lettera minuscola o un numero.

   Una volta creato il bucket, non è possibile modificarne il nome. Per ulteriori informazioni sulla denominazione dei bucket, consulta Regole di denominazione dei bucket.

> **Important**
>
> Evitare di includere informazioni riservate, ad esempio numeri di account, nel nome del bucket. Il nome bucket è visibile nell'URL che punta agli oggetti nel bucket.

5. In Outpost, seleziona l'Outpost in cui desideri sia ospitato il bucket.
6. In Bucket Versioning (Controllo delle versioni del bucket), imposta lo stato del controllo delle versioni S3 per il bucket S3 su Outposts su una delle seguenti opzioni:

   - Disable (Disabilita) (impostazione predefinita): il bucket rimane senza versione.
   - Enable (Abilita): il controllo delle versioni S3 è abilitato per gli oggetti nel bucket. A tutti gli oggetti aggiunti al bucket viene assegnato un ID versione univoco.

   Per ulteriori informazioni sulla funzione Controllo delle versioni S3, consulta Gestione del controllo delle versioni S3 per il bucket S3 su Outposts.
7. (Opzionale) Aggiungi eventuali tag facoltativi che desideri associare al bucket su Outposts. Puoi utilizzare i tag per monitorare i criteri per singoli progetti o gruppi di progetti, o per etichettare i bucket mediante i tag di allocazione dei costi.

   Per impostazione predefinita, tutti gli oggetti archiviati nel bucket su Outposts vengono memorizzati utilizzando la crittografia lato server con chiavi di crittografia gestite da Amazon S3 (SSE-S3). Puoi inoltre scegliere di archiviare esplicitamente gli oggetti utilizzando la crittografia lato server con chiavi di crittografia fornite dal cliente (SSE-C). Per modificare il tipo di crittografia, è necessario utilizzare l'API REST, AWS Command Line Interface(AWS CLI) oppure gli SDK AWS.
8. Nella sezione Impostazioni punto di accesso Outposts specifica il nome del punto di accesso.

   I punti di accesso S3 su Outposts semplificano la gestione dell'accesso ai dati su vasta scala per set di dati condivisi in S3 su Outposts. I punti di accesso sono endpoint di rete denominati che vengono collegati a bucket Outposts che puoi usare per eseguire operazioni su oggetti S3. Per ulteriori informazioni, consulta Access point.

   I nomi dei punti di accesso devono essere univoci all'interno dell'account per la regione e l'outpost e devono rispettare il Restrizioni e limitazioni degli access point..
9. Scegli il VPC per questo access point Amazon S3 su Outposts.

Se non hai un VPC scegli Create VPC (Crea VPC). Per ulteriori informazioni, consulta Creazione di access point limitati a un cloud privato virtuale.

Un Virtual Private Cloud (VPC) consente di avviare risorse AWS in una rete virtuale definita dall'utente. Questa rete virtuale è simile a una comune rete da gestire all'interno del proprio data center, ma con i vantaggi dell'impiego dell'infrastruttura scalabile di AWS.

10. (Facoltativo per un VPC esistente) Scegli una Endpoint subnet (Subnet endpoint) per l'endpoint.

    Una sottorete è un intervallo di indirizzi IP nel VPC. Se non disponi della sottorete desiderata, seleziona Crea sottorete. Per ulteriori informazioni, consulta Reti per S3 su Outposts.

11. (Facoltativo per un VPC esistente) Scegli una Endpoint security group (Gruppo di sicurezza endpoint) per l'endpoint.

    Un gruppo di sicurezza agisce da firewall virtuale per controllare il traffico in entrata e in uscita.

12. (Facoltativo per un VPC esistente) Scegli il Endpoint access type (Tipo di accesso all'endpoint):

    - Private (Privato): da utilizzare con il VPC.
    - Customer owned IP (IP di proprietà del cliente): da utilizzare con un pool di indirizzi IP di proprietà del cliente all'interno della rete On-Premise.

13. (Facoltativo) Specifica la policy del punto di accesso Outpost. La console visualizza automaticamente l'ARN (Amazon Resource Name del punto di accesso che può essere utilizzato nella policy.
14. Seleziona Crea bucket Outposts.

> **Note**
>
> Per la creazione dell'endpoint per gli outpost e perché il bucket sia pronto all'uso possono essere necessari fino a 5 minuti. Per configurare ulteriori impostazioni di bucket, seleziona Visualizza dettagli.

Fasi successive

Con Amazon S3 su Outposts, i dati degli oggetti vengono sempre archiviati nell'Outpost. Quando AWS installa un rack Outpost, i tuoi dati rimangono locali nel tuo Outpost per soddisfare i requisiti di residenza dei dati. I tuoi oggetti non lasciano mai il tuo Outpost e non sono in una Regione AWS. Dal momento che la AWS Management Console è ospitata nella regione, non puoi utilizzare la console

per caricare o gestire oggetti nel tuo Outpost. Tuttavia, puoi utilizzare l'API REST, AWS Command Line Interface (AWS CLI) e gli SDK AWS per caricare e gestire gli oggetti tramite i punti di accesso.

Dopo aver creato un bucket S3 on Outposts, un punto di accesso e un endpoint, puoi utilizzare AWS CLI o SDK per Java per caricare un oggetto nel bucket. Per ulteriori informazioni, consulta Carica un oggetto in un bucket S3 on Outposts.

Guida introduttiva all'utilizzo di AWS CLI and SDK for Java

Con Amazon S3 on Outposts, puoi creare bucket S3 sui tuoi AWS Outposts e archiviare e recuperare facilmente oggetti in locale per applicazioni che richiedono l'accesso locale ai dati, l'elaborazione locale dei dati e la residenza dei dati. S3 on Outposts offre una nuova classe di storage, S3 Outposts `OUTPOSTS` (), che utilizza le API di Amazon S3 ed è progettata per archiviare i dati in modo duraturo e ridondante su più dispositivi e server sul tuo. AWS Outposts Comunichi con il bucket Outpost utilizzando un punto di accesso e una connessione di endpoint su un Virtual Private Cloud (VPC). Sui bucket Outposts puoi utilizzare le stesse API e caratteristiche di Amazon S3, comprese policy di accesso, crittografia e tagging. Puoi usare S3 su Outposts tramite AWS Management Console AWS Command Line Interface ,AWS CLI() AWS , SDK o API REST. Per ulteriori informazioni, consulta Che cos'è Amazon S3 su Outposts?

Per iniziare a utilizzare S3 su Outposts devi creare un bucket, un punto di accesso e un endpoint. Quindi puoi caricare gli oggetti nel bucket. Gli esempi seguenti mostrano come iniziare a usare S3 su Outposts utilizzando AWS CLI l'SDK for Java. Per le nozioni di base sulla console, consulta Nozioni di base per l'utilizzo di AWS Management Console.

Argomenti

- Fase 1: creazione di un bucket
- Fase 3: creazione di un punto di accesso
- Fase 3: creazione di un endpoint
- Fase 4: caricamento di un oggetto in un bucket S3 su Outposts

Fase 1: creazione di un bucket

Gli esempi seguenti AWS CLI e quelli di SDK for Java mostrano come creare un bucket S3 on Outposts.

AWS CLI

Example

L'esempio seguente crea un bucket S3 su Outposts (`s3-outposts:CreateBucket`) utilizzando la AWS CLI. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
aws s3control create-bucket --bucket example-outposts-bucket --outpost-
id op-01ac5d28a6a232904
```

SDK for Java

Example

L'esempio seguente crea un bucket S3 su Outposts (`s3-outposts:CreateBucket`) utilizzando l'SDK per 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();

}
```

Fase 3: creazione di un punto di accesso

Per accedere al bucket Amazon S3 su Outposts devi creare e configurare un punto di accesso. Questi esempi mostrano come creare un punto di accesso utilizzando AWS CLI e l'SDK for Java.

Gli Access Point semplificano la gestione dell'accesso ai dati su vasta scala per set di dati condivisi in Amazon S3. Gli access point sono endpoint di rete denominati che vengono collegati a bucket che possono essere utilizzati per eseguire operazioni su oggetti di Amazon S3, ad esempio `GetObject` e `PutObject`. Con S3 su Outposts devi utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outposts. I punti di accesso supportano solo l' virtual-host-style indirizzamento.

AWS CLI

Example

L' AWS CLI esempio seguente crea un punto di accesso per un bucket Outposts. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
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

Nell'esempio SDK per Java seguente viene creato un punto di accesso per un bucket Outposts. Per utilizzare questo comando, sostituisci *`user input placeholders`* con le tue informazioni.

```
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();

}
```

Fase 3: creazione di un endpoint

Per instradare le richieste verso un punto di accesso Amazon S3 su Outposts, è necessario creare e configurare un endpoint S3 su Outposts. Per creare un endpoint, è necessario disporre di una connessione attiva con il collegamento del servizio alla regione di origine degli outpost. Ogni cloud privato virtuale (VPC) del tuo Outpost può avere un endpoint associato. Per ulteriori informazioni sull'endpoint, consulta Requisiti di rete di S3 su Outposts. È necessario creare un endpoint per poter accedere ai bucket Outposts ed eseguire operazioni sugli oggetti. Per ulteriori informazioni, consulta Endpoint.

Questi esempi mostrano come creare un endpoint utilizzando AWS CLI e l'SDK for Java. Per ulteriori informazioni sulle autorizzazioni richieste per la creazione e la gestione degli endpoint, consulta Autorizzazioni per endpoint S3 su Outposts.

AWS CLI

Example

L' AWS CLI esempio seguente crea un endpoint per un Outpost utilizzando il tipo di accesso alle risorse VPC. Il VPC deriva dalla sottorete. Per eseguire questo comando, sostituire *user input placeholders* con le proprie informazioni.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
 subnet-8c7a57c5 --security-group-id sg-ab19e0d1
```

L' AWS CLI esempio seguente crea un endpoint per un Outpost utilizzando il tipo di accesso del pool di indirizzi IP (pool CoIP) di proprietà del cliente. Per eseguire questo comando, sostituire *user input placeholders* con le proprie informazioni.

```
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

Nell'esempio SDK per Java seguente viene creato un endpoint per un outpost. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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());
}
```

Fase 4: caricamento di un oggetto in un bucket S3 su Outposts

Per caricare un oggetto, consulta Carica un oggetto in un bucket S3 on Outposts.

Reti per S3 su Outposts

Puoi utilizzare S3 su Outposts per archiviare e recuperare oggetti On-Premise per le applicazioni che richiedono l'accesso ai dati locali, l'elaborazione dei dati e la residenza dei dati. Questa sezione descrive i requisiti di rete per l'accesso a S3 su Outposts.

Argomenti

- Scelta del tipo di accesso di rete
- Accesso a bucket e oggetti S3 su Outposts
- Interfacce di rete elastiche tra account

Scelta del tipo di accesso di rete

Puoi accedere a S3 su Outposts dall'interno di un VPC o dalla tua rete on-premise. Comunichi con il bucket Outpost utilizzando un punto di accesso e una connessione di endpoint. In tal modo il traffico tra il tuo VPC e i bucket S3 on Outposts viene mantenuto all'interno della rete AWS. Quando crei un endpoint, devi specificare il tipo di accesso all'endpoint tra `Private` (per l'instradamento al VPC) e `CustomerOwnedIp` (per il pool di indirizzi IP di proprietà del cliente [pool CoIP]).

- `Private` (per l'instradamento al VPC). Se non indichi il tipo di accesso, S3 su Outposts utilizza `Private` per impostazione predefinita. Con il tipo di accesso `Private`, le istanze nel VPC non richiedono indirizzi IP pubblici per comunicare con le risorse nell'Outpost. Puoi lavorare con S3 su Outposts da un VPC. Questo tipo di endpoint è accessibile dalla rete on-premise attraverso il routing VPC diretto. Per ulteriori informazioni, consulta Local gateway route tables nella Guida per l'utente di AWS Outposts.
- `CustomerOwnedIp` (per il pool CoIP). Se non usi il valore predefinito tipo di accesso `Private` e scegli `CustomerOwnedIp`, devi specificare un intervallo di indirizzi IP. Puoi utilizzare questo tipo di accesso per lavorare con S3 su Outposts dalla rete On-Premise e in un VPC. Quando accedi a S3 su Outposts all'interno di un VPC, il traffico è limitato alla larghezza di banda del gateway locale.

Accesso a bucket e oggetti S3 su Outposts

Per accedere ai tuoi oggetti e ai bucket di S3 su Outposts, devi disporre di:

- Un punto di accesso per il VPC.
- Un endpoint per lo stesso VPC.
- Una connessione attiva tra il tuo Outpost e la tua Regione AWS. Per ulteriori informazioni su come connettere l'outpost a una regione, consulta Outpost connectivity to AWS Regions nella Guida per l'utente di AWS Outposts.

Per ulteriori informazioni sull'accesso a bucket e oggetti in S3 su Outposts, consulta Utilizzo di bucket S3 su Outposts e Utilizzo di oggetti S3 su Outposts.

Interfacce di rete elastiche tra account

Gli endpoint S3 su Outposts sono risorse denominate con ARN (Amazon Resource Name). Quando vengono creati questi endpoint, AWS Outposts imposta quattro interfacce di rete elastiche tra

account. Le interfacce di rete elastiche tra account S3 su Outposts sono come altre interfacce di rete con una sola eccezione: S3 on Outposts associa le interfacce di rete elastiche tra account alle istanze Amazon EC2.

Il DNS (Domain Name System) di S3 su Outposts userà il bilanciamento del carico delle tue richieste sull'interfaccia di rete elastica tra account. S3 su Outposts crea l'interfaccia di rete elastica tra account nel tuo account AWS, visibile dal riquadro Interfacce di rete della console Amazon EC2.

Per gli endpoint che utilizzano il tipo di accesso al pool CoIP, S3 su Outposts alloca e associa gli indirizzi IP all'interfaccia di rete elastica tra account dal pool CoIP configurato.

Utilizzo di bucket S3 su Outposts

Con Amazon S3 su Outposts è possibile creare bucket S3 su AWS Outposts, nonché archiviare e recuperare facilmente gli oggetti On-Premise per le applicazioni che richiedono l'accesso ai dati in locale, l'elaborazione dei dati in locale e la residenza dei dati. S3 su Outposts fornisce una nuova classe di storage, S3 Outposts (OUTPOSTS), che utilizza le API Amazon S3 ed è progettata per archiviare i dati in modo durevole e ridondante su più dispositivi e server su AWS Outposts. Sui bucket Outposts puoi utilizzare le stesse API e caratteristiche di Amazon S3, comprese policy di accesso, crittografia e tagging. Per ulteriori informazioni, consulta Che cos'è Amazon S3 su Outposts?

Comunichi con i bucket Outpost utilizzando un punto di accesso e una connessione di endpoint su un Virtual Private Cloud (VPC). Per accedere ai tuoi oggetti e bucket S3 su Outposts, devi disporre di un punto di accesso per il VPC e di un endpoint per lo stesso VPC. Per ulteriori informazioni, consulta Reti per S3 su Outposts.

Bucket

In S3 su Outposts, i nomi dei bucket sono univoci in un Outpost e richiedono il codice Regione AWS per la regione in cui è presente Outpost, l'ID Account AWS, l'ID Outpost e il nome del bucket per identificarli.

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/bucket/bucket-name
```

Per ulteriori informazioni, consulta ARN delle risorse per S3 su Outposts.

Access point

Amazon S3 su Outposts supporta i punti di accesso configurati solo per i virtual private cloud (VPC) come unico mezzo per accedere ai bucket di Outposts.

Gli Access Point semplificano la gestione dell'accesso ai dati su vasta scala per set di dati condivisi in Amazon S3. Gli access point sono endpoint di rete denominati che vengono collegati a bucket che possono essere utilizzati per eseguire operazioni su oggetti di Amazon S3, ad esempio `GetObject` e `PutObject`. Con S3 su Outposts devi utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outposts. I punti di accesso supportano solo l'indirizzamento in stile hosting virtuale.

Il seguente esempio illustra il formato ARN da utilizzare per i punti di accesso di S3 su Outposts. L'ARN del punto di accesso il codice Regione AWS per la regione in cui è presente l'Outpost, l'ID Account AWS, l'ID Outpost e il nome del punto di accesso.

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Endpoint

Per instradare le richieste a un punto di accesso S3 su Outposts, è necessario creare e configurare un endpoint S3 su Outposts. Con gli endpoint S3 su Outposts, puoi collegare privatamente il tuo VPC al tuo bucket Outpost. Gli endpoint S3 su Outposts sono URI (Uniform Resource Identifier) virtuali del punto di ingresso al bucket S3 su Outposts. Sono componenti VPC con scalabilità orizzontale, ridondanza e disponibilità elevata.

Ogni cloud privato virtuale (VPC) del tuo Outpost può avere un endpoint associato, per un massimo di 100 endpoint per Outpost. È necessario creare questi endpoint per poter accedere ai bucket Outpost ed eseguire operazioni sugli oggetti. La creazione di questi endpoint consente inoltre che il modello API e i comportamenti siano gli stessi consentendo alle stesse operazioni di funzionare in S3 e S3 su Outposts.

Operazioni API in S3 su Outposts

S3 su Outposts ospita un endpoint separato diverso dall'endpoint Amazon S3 per gestire le operazioni API del bucket Outpost. Questo endpoint è `s3-outposts.region.amazonaws.com`.

Per utilizzare le operazioni API Amazon S3, è necessario firmare il bucket e gli oggetti utilizzando il formato ARN corretto. Gli ARN per le operazioni API vanno inoltrati in modo che Amazon S3 possa determinare se la richiesta è per Amazon S3 (`s3-control.region.amazonaws.com`) o per S3 su

Outposts (`s3-outposts.`*`region`*`.amazonaws.com`). In base al formato dell'ARN, S3 può quindi firmare e instradare la richiesta in modo appropriato.

Ogni volta che la richiesta viene inviata al pannello di controllo Amazon S3, l'SDK estrae i componenti dall'ARN e include l'intestazione aggiuntiva `x-amz-`*`outpost-id`* con il valore *`outpost-id`* estratto dall'ARN. Il nome del servizio dall'ARN verrà utilizzato per firmare la richiesta prima che venga instradata all'endpoint S3 su Outposts. Questo comportamento si applica a tutte le operazioni API gestite dal client `s3control`.

Nella tabella seguente sono riportate le operazioni API estese per Amazon S3 su Outposts e le loro modifiche rispetto ad Amazon S3.

API	S3 sul valore del parametro Outposts	
`CreateBucket`	Nome bucket come ARN, ID Outpost	
`ListRegionalBuckets`	ID Outpost	
`DeleteBucket`	Nome del bucket come ARN	
`DeleteBucketLifecycleConfiguration`	Nome del bucket come ARN	
`GetBucketLifecycleConfiguration`	Nome del bucket come ARN	
`PutBucketLifecycleConfiguration`	Nome del bucket come ARN	
`GetBucketPolicy`	Nome del bucket come ARN	
`PutBucketPolicy`	Nome del bucket come ARN	
`DeleteBucketPolicy`	Nome del bucket come ARN	
`GetBucketTagging`	Nome del bucket come ARN	
`PutBucketTagging`	Nome del bucket come ARN	

API	S3 sul valore del parametro Outposts	
`DeleteBucketTagging`	Nome del bucket come ARN	
`CreateAccessPoint`	Nome del punto di accesso come ARN	
`DeleteAccessPoint`	Nome del punto di accesso come ARN	
`GetAccessPoint`	Nome del punto di accesso come ARN	
`GetAccessPoint`	Nome del punto di accesso come ARN	
`ListAccessPoints`	Nome del punto di accesso come ARN	
`PutAccessPointPolicy`	Nome del punto di accesso come ARN	
`GetAccessPointPolicy`	Nome del punto di accesso come ARN	
`DeleteAccessPointPolicy`	Nome del punto di accesso come ARN	

Creazione e gestione di bucket S3 su Outposts

Per ulteriori informazioni sulla creazione e sulla gestione dei bucket S3 su Outposts, consulta i seguenti argomenti.

Argomenti

- Creazione di un bucket S3 su Outposts
- Aggiunta di tag per bucket S3 su Outposts
- Gestione dell'accesso al bucket Amazon S3 su Outposts utilizzando la policy del bucket

- Elenco di bucket Amazon S3 su Outposts
- Recupero di un bucket S3 su Outposts utilizzando AWS CLI e SDK per Java
- Eliminazione del bucket Amazon S3 su Outposts
- Utilizzo dei punti di accesso Amazon S3 su Outposts
- Utilizzo degli endpoint Amazon S3 su Outposts

Creazione di un bucket S3 su Outposts

Con Amazon S3 su Outposts è possibile creare bucket S3 su AWS Outposts, nonché archiviare e recuperare facilmente gli oggetti on-Premise per le applicazioni che richiedono l'accesso ai dati in locale, l'elaborazione dei dati in locale e la residenza dei dati. S3 su Outposts fornisce una nuova classe di archiviazione, S3 Outposts (`OUTPOSTS`), che utilizza le API Amazon S3 ed è progettata per archiviare i dati in modo durevole e ridondante su più dispositivi e server su AWS Outposts. Comunichi con il bucket Outpost utilizzando un punto di accesso e una connessione di endpoint su un Virtual Private Cloud (VPC). Sui bucket Outposts puoi utilizzare le stesse API e caratteristiche di Amazon S3, comprese policy di accesso, crittografia e tagging. Puoi utilizzare S3 su Outposts tramite la AWS Management Console, AWS Command Line Interface (AWS CLI), gli SDK AWS o l'API REST. Per ulteriori informazioni, consulta Che cos'è Amazon S3 su Outposts?.

> **Note**
>
> L'Account AWS che crea il bucket lo possiede ed è l'unico che può eseguire azioni su di esso. I bucket dispongono di proprietà di configurazione come Outpost, tag, crittografia di default e impostazioni del punto di accesso. Le impostazioni del punto di accesso includono il Virtual Private Cloud (VPC), la policy del punto di accesso per l'accesso agli oggetti nel bucket e altri metadati. Per ulteriori informazioni, consulta Specifiche di S3 su Outposts.
> Se desideri creare un bucket che utilizza AWS PrivateLink per fornire l'accesso alla gestione di bucket ed endpoint tramite endpoint VPC dell'interfaccia nel cloud privato virtuale (VPC), consulta AWS PrivateLink per S3 su Outposts.

Gli esempi seguenti illustrano come creare un bucket S3 su Outposts utilizzando la AWS Management Console, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Seleziona Crea bucket Outposts.
4. In Bucket name (Nome bucket), immettere un nome conforme a DNS (Domain Name System) per il bucket.

   Il nome del bucket deve:

   - Essere univoco nell'Account AWS, nell'Outpost e nella Regione AWS in cui si trova l'Outpost.
   - Contenere da 3 a 63 caratteri.
   - Non contiene caratteri maiuscoli.
   - Iniziare con una lettera minuscola o un numero.

   Una volta creato il bucket, non è possibile modificarne il nome. Per ulteriori informazioni sulla denominazione dei bucket, consulta Regole di denominazione dei bucket.

   > **Important**
   >
   > Evitare di includere informazioni riservate, ad esempio numeri di account, nel nome del bucket. Il nome bucket è visibile nell'URL che punta agli oggetti nel bucket.

5. In Outpost, seleziona l'Outpost in cui desideri sia ospitato il bucket.
6. In Bucket Versioning (Controllo delle versioni del bucket), imposta lo stato del controllo delle versioni S3 per il bucket S3 su Outposts su una delle seguenti opzioni:

   - Disable (Disabilita) (impostazione predefinita): il bucket rimane senza versione.
   - Enable (Abilita): il controllo delle versioni S3 è abilitato per gli oggetti nel bucket. A tutti gli oggetti aggiunti al bucket viene assegnato un ID versione univoco.

   Per ulteriori informazioni sulla funzione Controllo delle versioni S3, consulta Gestione del controllo delle versioni S3 per il bucket S3 su Outposts.

7. (Opzionale) Aggiungi eventuali tag facoltativi che desideri associare al bucket su Outposts. Puoi utilizzare i tag per monitorare i criteri per singoli progetti o gruppi di progetti, o per etichettare i bucket mediante i tag di allocazione dei costi.

   Per impostazione predefinita, tutti gli oggetti archiviati nel bucket su Outposts vengono memorizzati utilizzando la crittografia lato server con chiavi di crittografia gestite da Amazon S3 (SSE-S3). Puoi inoltre scegliere di archiviare esplicitamente gli oggetti utilizzando la crittografia lato server con chiavi di crittografia fornite dal cliente (SSE-C). Per modificare il tipo di crittografia, è necessario utilizzare l'API REST, AWS Command Line Interface(AWS CLI) oppure gli SDK AWS.

8. Nella sezione Impostazioni punto di accesso Outposts specifica il nome del punto di accesso.

   I punti di accesso S3 su Outposts semplificano la gestione dell'accesso ai dati su vasta scala per set di dati condivisi in S3 su Outposts. I punti di accesso sono endpoint di rete denominati che vengono collegati a bucket Outposts che puoi usare per eseguire operazioni su oggetti S3. Per ulteriori informazioni, consulta Access point.

   I nomi dei punti di accesso devono essere univoci all'interno dell'account per la regione e l'outpost e devono rispettare il Restrizioni e limitazioni degli access point..

9. Scegli il VPC per questo access point Amazon S3 su Outposts.

   Se non hai un VPC scegli Create VPC (Crea VPC). Per ulteriori informazioni, consulta Creazione di access point limitati a un cloud privato virtuale.

   Un Virtual Private Cloud (VPC) consente di avviare risorse AWS in una rete virtuale definita dall'utente. Questa rete virtuale è simile a una comune rete da gestire all'interno del proprio data center, ma con i vantaggi dell'impiego dell'infrastruttura scalabile di AWS.

10. (Facoltativo per un VPC esistente) Scegli una Endpoint subnet (Subnet endpoint) per l'endpoint.

    Una sottorete è un intervallo di indirizzi IP nel VPC. Se non disponi della sottorete desiderata, seleziona Crea sottorete. Per ulteriori informazioni, consulta Reti per S3 su Outposts.

11. (Facoltativo per un VPC esistente) Scegli una Endpoint security group (Gruppo di sicurezza endpoint) per l'endpoint.

    Un gruppo di sicurezza agisce da firewall virtuale per controllare il traffico in entrata e in uscita.

12. (Facoltativo per un VPC esistente) Scegli il Endpoint access type (Tipo di accesso all'endpoint):

    - Private (Privato): da utilizzare con il VPC.

- Customer owned IP (IP di proprietà del cliente): da utilizzare con un pool di indirizzi IP di proprietà del cliente all'interno della rete On-Premise.

13. (Facoltativo) Specifica la policy del punto di accesso Outpost. La console visualizza automaticamente l'ARN (Amazon Resource Name del punto di accesso che può essere utilizzato nella policy.
14. Seleziona Crea bucket Outposts.

> **Note**
>
> Per la creazione dell'endpoint per gli outpost e perché il bucket sia pronto all'uso possono essere necessari fino a 5 minuti. Per configurare ulteriori impostazioni di bucket, seleziona Visualizza dettagli.

Utilizzo di AWS CLI

Example

L'esempio seguente crea un bucket S3 su Outposts (`s3-outposts:CreateBucket`) utilizzando la AWS CLI. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
aws s3control create-bucket --bucket example-outposts-bucket --outpost-
id op-01ac5d28a6a232904
```

Utilizzo dell'SDK AWS per Java

Example

L'esempio seguente crea un bucket S3 su Outposts (`s3-outposts:CreateBucket`) utilizzando l'SDK per 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();

}
```

Aggiunta di tag per bucket S3 su Outposts

Puoi aggiungere tag per i bucket Amazon S3 su Outposts per tenere traccia dei costi di storage o di altri criteri per singoli progetti o gruppi di progetti.

> **Note**
>
> L'Account AWS che crea il bucket lo possiede ed è l'unico che ne può cambiare i tag.

Utilizzo della console S3

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Seleziona il bucket Outposts di cui desideri modificare i tag.
4. Scegliere la scheda Properties (Proprietà).
5. In Tags, scegliere Edit (Modifica).
6. Scegli Add new tag (Aggiungi nuovo tag) e completa i campi Key (Chiave) e facoltativamente Value (Valore).

   Aggiungi eventuali tag da associare al bucket Outposts per tenere traccia ddi altri criteri per singoli progetti o gruppi di progetti.
7. Scegliere Save changes (Salva modifiche).

Tramite AWS CLI

Il seguente esempio AWS CLI applica una configurazione di tag a un bucket S3 su Outposts utilizzando un documento JSON nella cartella corrente che specifica i tag (*tagging.json*). Per utilizzare questo comando, sostituisci *user input placeholder* con le tue informazioni.

```
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"
     }
   ]
}
```

Il seguente esempio AWS CLI applica una configurazione di tag a un bucket S3 su Outposts direttamente dalla riga di comando.

```
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}]'
```

Per ulteriori informazioni su questo comando, consulta put-bucket-taggingnella Guida di riferimento di AWS CLI.

Gestione dell'accesso al bucket Amazon S3 su Outposts utilizzando la policy del bucket

Una policy di bucket è una policy AWS Identity and Access Management (IAM) basata su risorse che puoi utilizzare per concedere autorizzazioni di accesso al bucket e agli oggetti che contiene. Solo il proprietario del bucket può associare una policy a un bucket. Le autorizzazioni allegate a un bucket si applicano a tutti gli oggetti del bucket di proprietà del proprietario del bucket. Le policy di bucket sono limitate a una dimensione di 20 KB. Per ulteriori informazioni, consulta Policy del bucket.

Puoi aggiornare la policy del bucket per gestire l'accesso al bucket Amazon S3 su Outposts. Per ulteriori informazioni, consulta i seguenti argomenti.

Argomenti

Aggiunta o modifica di una policy di un bucket Amazon S3 su Outposts

Una policy di bucket è una policy AWS Identity and Access Management (IAM) basata su risorse che puoi utilizzare per concedere autorizzazioni di accesso al bucket e agli oggetti che contiene. Solo il proprietario del bucket può associare una policy a un bucket. Le autorizzazioni allegate a un bucket si applicano a tutti gli oggetti del bucket di proprietà del proprietario del bucket. Le policy di bucket sono limitate a una dimensione di 20 KB. Per ulteriori informazioni, consultare Policy del bucket.

I seguenti argomenti illustrano come aggiornare la policy del bucket Amazon S3 su Outposts utilizzando la AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS SDK for Java.

Utilizzo della console S3

Per creare o modificare una policy di bucket

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Seleziona il bucket Outposts di cui desideri modificare la policy.
4. Scegliere la scheda Permissions (Autorizzazioni).
5. Nella sezione Outposts bucket policy (Policy del bucket Outposts) scegliEdit (Modifica) per creare o modificare la policy.

   Ora puoi decidere di aggiungere o modificare la policy del bucket S3 su Outposts. Per ulteriori informazioni, consultare Configurazione di IAM con S3 su Outposts.

Utilizzo di AWS CLI

Nell'esempio della AWS CLI seguente viene inserita una policy su un bucket Outposts.

1. Salva la seguente policy di bucket in un file JSON. In questo esempio, il file è denominato `policy1.json`. Sostituire *`user input placeholders`* con le proprie informazioni.

```
{
   "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. Inviare il file JSON come parte del comando CLI `put-bucket-policy`. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
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
```

Utilizzo dell'SDK AWS per Java

Nell'esempio SDK per Java seguente viene inserita una policy su un bucket 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());

}
```

Visualizzazione della policy del bucket Amazon S3 su Outposts

Una policy di bucket è una policy AWS Identity and Access Management (IAM) basata su risorse che puoi utilizzare per concedere autorizzazioni di accesso al bucket e agli oggetti che contiene. Solo il proprietario del bucket può associare una policy a un bucket. Le autorizzazioni allegate a un bucket si applicano a tutti gli oggetti del bucket di proprietà del proprietario del bucket. Le policy di bucket sono limitate a una dimensione di 20 KB. Per ulteriori informazioni, consultare Policy del bucket.

I seguenti argomenti illustrano come visualizzare la policy del bucket Amazon S3 su Outposts utilizzando la AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS SDK for Java.

Utilizzo della console S3

Per creare o modificare una policy di bucket

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Seleziona il bucket Outposts di cui desideri modificare l'autorizzazione.
4. Scegliere la scheda Permissions (Autorizzazioni).
5. Nella sezione Outposts bucket policy (Policy del bucket Outposts) puoi rivedere la policy del bucket esistente. Per ulteriori informazioni, consulta Configurazione di IAM con S3 su Outposts.

Utilizzo di AWS CLI

Nell'esempio della AWS CLI seguente si ottiene la policy per un bucket di Outposts. Per eseguire questo comando, sostituire *user input placeholders* con le proprie informazioni.

```
aws s3control get-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket
```

Utilizzo dell'SDK AWS per Java

Nell'esempio SDK per Java seguente si ottiene una policy per un bucket 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());

}
```

Eliminazione della policy del bucket Amazon S3 su Outposts

Una policy di bucket è una policy AWS Identity and Access Management (IAM) basata su risorse che puoi utilizzare per concedere autorizzazioni di accesso al bucket e agli oggetti che contiene. Solo il proprietario del bucket può associare una policy a un bucket. Le autorizzazioni allegate a un bucket si applicano a tutti gli oggetti del bucket di proprietà del proprietario del bucket. Le policy di bucket sono limitate a una dimensione di 20 KB. Per ulteriori informazioni, consultare Policy del bucket.

I seguenti argomenti illustrano come visualizzare la policy del bucket Amazon S3 su Outposts utilizzando la AWS Management Console o AWS Command Line Interface (AWS CLI).

Utilizzo della console S3

Eliminare una policy di bucket

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Seleziona il bucket Outposts di cui desideri modificare l'autorizzazione.
4. Scegliere la scheda Permissions (Autorizzazioni).
5. Nella sezione Policy bucket Outposts, seleziona Elimina.
6. Confermare l'eliminazione.

Tramite AWS CLI

Nell'esempio seguente viene eliminata la policy del bucket S3 su Outposts (`s3-outposts:DeleteBucket`) utilizzando AWS CLI. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
aws s3control delete-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket
```

Esempi di policy di bucket

Con le policy relative ai bucket S3 on Outposts, puoi proteggere l'accesso agli oggetti nei bucket S3 on Outposts, in modo che solo gli utenti con le autorizzazioni appropriate possano accedervi. Puoi anche impedire agli utenti autenticati senza le autorizzazioni appropriate di accedere alle tue risorse S3 on Outposts.

Questa sezione presenta esempi di casi d'uso tipici per le policy dei bucket di S3 on Outposts. Per testare queste policy, sostituisci *`user input placeholders`* con le tue informazioni (come il nome del bucket).

Per concedere o negare le autorizzazioni per un insieme di oggetti, puoi utilizzare caratteri jolly (`*`) nei nomi delle risorse Amazon (ARN) e altri valori. Ad esempio, puoi controllare l'accesso a gruppi di oggetti che iniziano con un prefisso comune o terminano con una determinata estensione, come `.html`.

Per informazioni sul linguaggio delle policy AWS Identity and Access Management (IAM), consulta Configurazione di IAM con S3 su Outposts.

> **Note**
>
> Quando si testano s3outpostsle autorizzazioni utilizzando la console Amazon S3, è necessario concedere le autorizzazioni aggiuntive richieste dalla console, ad esempio, e `s3outposts:createendpoint` così `s3outposts:listendpoints` via.

Risorse aggiuntive per la creazione di policy bucket

- Per un elenco delle azioni, delle risorse e delle chiavi di condizione IAM che puoi utilizzare per creare una policy sui bucket S3 on Outposts, consulta Azioni, risorse e chiavi di condizione per Amazon S3 on Outposts.
- Per indicazioni su come creare la tua policy S3 on Outposts, consulta. Aggiunta o modifica di una policy di un bucket Amazon S3 su Outposts

Argomenti

- Gestione dell'accesso a un bucket Amazon S3 on Outposts in base a indirizzi IP specifici

Gestione dell'accesso a un bucket Amazon S3 on Outposts in base a indirizzi IP specifici

Una policy di bucket è una policy AWS Identity and Access Management (IAM) basata su risorse che puoi utilizzare per concedere autorizzazioni di accesso al bucket e agli oggetti che contiene. Solo il proprietario del bucket può associare una policy a un bucket. Le autorizzazioni allegate a un bucket si applicano a tutti gli oggetti del bucket di proprietà del proprietario del bucket. Le policy di bucket sono limitate a una dimensione di 20 KB. Per ulteriori informazioni, consulta Policy del bucket.

Limitare l'accesso a indirizzi IP specifici

L'esempio seguente impedisce a tutti gli utenti di eseguire qualsiasi operazione di S3 on Outposts sugli oggetti nei bucket specificati a meno che la richiesta non provenga dall'intervallo di indirizzi IP specificato.

> **Note**
>
> Quando limiti l'accesso a un indirizzo IP specifico, assicurati di specificare anche quali endpoint VPC, indirizzi IP di origine VPC o indirizzi IP esterni possono accedere al bucket S3 on Outposts. Altrimenti, potresti perdere l'accesso al bucket se la tua politica impedisce

a tutti gli utenti di eseguire qualsiasi s3outpostsoperazione sugli oggetti nel tuo bucket S3 on Outposts senza le autorizzazioni appropriate già presenti.

La `Condition` dichiarazione di questa politica si identifica *`192.0.2.0/24`*come l'intervallo di indirizzi IP versione 4 (IPv4) consentiti.

Il blocco `Condition` utilizza la condizione `NotIpAddress` e la chiave di condizione `aws:SourceIp`, che è una chiave di condizione valida per tutto AWS. La chiave di condizione `aws:SourceIp` può essere utilizzata solo per intervalli di indirizzi IP pubblici. Per ulteriori informazioni su queste chiavi di condizione, consulta Azioni, risorse e chiavi di condizione per S3 on Outposts. I valori IPv4 `aws:SourceIp` utilizzano la notazione CIDR standard. Per ulteriori informazioni, consulta il riferimento agli elementi delle policy IAM JSON nella IAM User Guide.

Warning

Prima di utilizzare questa policy di S3 on Outposts, sostituisci *`192.0.2.0/24`*l'intervallo di indirizzi IP in questo esempio con un valore appropriato per il tuo caso d'uso. Altrimenti, perderai la possibilità di accedere al tuo 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"
                }
```

```
            }
        }
    ]
}
```

Consentire entrambi gli indirizzi IPv4 e IPv6

Quando inizi a utilizzare gli indirizzi IPv6, è consigliabile aggiornare tutte le policy dell'organizzazione con gli intervalli di indirizzi IPv6 in aggiunta agli intervalli di indirizzi IPv4 esistenti. Ciò contribuirà a garantire che le politiche continuino a funzionare durante la transizione a IPv6.

Il seguente esempio di policy bucket di S3 on Outposts mostra come combinare intervalli di indirizzi IPv4 e IPv6 per coprire tutti gli indirizzi IP validi dell'organizzazione. La policy di esempio permette l'accesso agli indirizzi IP di esempio *192.0.2.1* e *2001:DB8:1234:5678::1* e lo nega agli indirizzi *203.0.113.1* e *2001:DB8:1234:5678:ABCD::1*.

La chiave di condizione `aws:SourceIp` può essere utilizzata solo per intervalli di indirizzi IP pubblici. I valori IPv6 per `aws:SourceIp` devono essere nel formato CIDR standard. Per IPv6 è supportato l'utilizzo di `::` per rappresentare un intervallo di zeri (0), ad esempio `2001:DB8:1234:5678::/64`. Per ulteriori informazioni, consulta Operatori di condizione con indirizzo IP nella Guida per l'utente di IAM.

> **Warning**
>
> Sostituisci gli intervalli di indirizzi IP in questo esempio con valori appropriati per il tuo caso d'uso prima di utilizzare questa politica S3 on Outposts. In caso contrario, si potrebbe perdere la possibilità di accedere al 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"
                    ]
                }
            }
        }
    ]
}
```

Elenco di bucket Amazon S3 su Outposts

Con Amazon S3 su Outposts è possibile creare bucket S3 su AWS Outposts, nonché archiviare e recuperare facilmente gli oggetti on-Premise per le applicazioni che richiedono l'accesso ai dati in locale, l'elaborazione dei dati in locale e la residenza dei dati. S3 su Outposts fornisce una nuova classe di storage, S3 Outposts (OUTPOSTS), che utilizza le API Amazon S3 ed è progettata per archiviare i dati in modo durevole e ridondante su più dispositivi e server su AWS Outposts. Comunichi con il bucket Outpost utilizzando un punto di accesso e una connessione di endpoint su un Virtual Private Cloud (VPC). Sui bucket Outposts puoi utilizzare le stesse API e caratteristiche di Amazon S3, comprese policy di accesso, crittografia e tagging. Puoi utilizzare S3 su Outposts tramite la AWS Management Console, AWS Command Line Interface (AWS CLI), gli SDK AWS o l'API REST. Per ulteriori informazioni, consulta Che cos'è Amazon S3 su Outposts?

Per ulteriori informazioni sull'utilizzo dei bucket in S3 su Outposts, consulta Utilizzo di bucket S3 su Outposts.

Gli esempi seguenti mostrano come ottenere l'elenco dei bucket S3 su Outposts utilizzando AWS Management Console, AWS CLI e AWS SDK for Java.

Utilizzo della console S3

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. In Outposts buckets (Bucket Outposts), consulta il tuo elenco di bucket S3 su Outposts.

Tramite AWS CLI

Nell'esempio AWS CLI seguente viene ottenuto un elenco di bucket in un Outpost. Per eseguire questo comando, sostituire *`user input placeholder`* con le proprie informazioni. Per ulteriori informazioni su questo comando, consulta list-regional-bucketsnella Guida di riferimento a AWS CLI.

```
aws s3control list-regional-buckets --account-id 123456789012 --outpost-
id op-01ac5d28a6a232904
```

Utilizzo dell'SDK AWS per Java

Nell'esempio SDK per Java seguente si ottiene un elenco di bucket in un outpost. Per ulteriori informazioni, consulta ListRegionalBuckets nella Documentazione di riferimento delle API di 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());

}
```

Recupero di un bucket S3 su Outposts utilizzando AWS CLI e SDK per Java

Con Amazon S3 su Outposts è possibile creare bucket S3 su AWS Outposts, nonché archiviare e recuperare facilmente gli oggetti on-Premise per le applicazioni che richiedono l'accesso ai dati in locale, l'elaborazione dei dati in locale e la residenza dei dati. S3 su Outposts fornisce una nuova classe di storage, S3 Outposts (OUTPOSTS), che utilizza le API Amazon S3 ed è progettata per archiviare i dati in modo durevole e ridondante su più dispositivi e server su AWS Outposts. Comunichi con il bucket Outpost utilizzando un punto di accesso e una connessione di endpoint su un Virtual Private Cloud (VPC). Sui bucket Outposts puoi utilizzare le stesse API e caratteristiche di Amazon S3, comprese policy di accesso, crittografia e tagging. Puoi utilizzare S3 su Outposts tramite la AWS Management Console, AWS Command Line Interface (AWS CLI), gli SDK AWS o l'API REST. Per ulteriori informazioni, consulta Che cos'è Amazon S3 su Outposts?

Gli esempi seguenti illustrano come recuperare un bucket S3 su Outposts utilizzando AWS CLI e AWS SDK for Java.

> **Note**
>
> Utilizzando questa operazione con Amazon S3 su Outposts tramite AWS CLI o gli SDK AWS, fornisci l'ARN del punto di accesso per Outposts invece del nome del bucket. Il punto di accesso ARN assume il seguente modulo, dove *region* è il codice Regione AWS per la Regione in cui ha sede Outpost:
> `arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point`
> Per ulteriori informazioni sugli ARN S3 su Outposts, consulta ARN delle risorse per S3 su Outposts.

Tramite AWS CLI

Nell'esempio S3 su Outposts seguente si ottiene un bucket utilizzando AWS CLI. Per usare questo comando, sostituire ogni *user input placeholder* con le proprie informazioni. Per ulteriori informazioni su questo comando, consulta get-bucket nella Guida di riferimento AWS CLI.

```
aws s3control get-bucket --account-id 123456789012 --bucket "arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket"
```

Utilizzo dell'SDK AWS per Java

Nell'esempio S3 su Outposts seguente viene ottenuto un bucket utilizzando SDK per Java. Per ulteriori informazioni, consulta la sezione GetBucket in Guida di riferimento all'API di 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());

}
```

Eliminazione del bucket Amazon S3 su Outposts

Con Amazon S3 su Outposts è possibile creare bucket S3 su AWS Outposts, nonché archiviare e recuperare facilmente gli oggetti on-Premise per le applicazioni che richiedono l'accesso ai dati in locale, l'elaborazione dei dati in locale e la residenza dei dati. S3 su Outposts fornisce una nuova classe di storage, S3 Outposts (OUTPOSTS), che utilizza le API Amazon S3 ed è progettata per archiviare i dati in modo durevole e ridondante su più dispositivi e server su AWS Outposts. Comunichi con il bucket Outpost utilizzando un punto di accesso e una connessione di endpoint su un Virtual Private Cloud (VPC). Sui bucket Outposts puoi utilizzare le stesse API e caratteristiche di Amazon S3, comprese policy di accesso, crittografia e tagging. Puoi utilizzare S3 su Outposts tramite la AWS Management Console, AWS Command Line Interface (AWS CLI), gli SDK AWS o l'API REST. Per ulteriori informazioni, consulta Che cos'è Amazon S3 su Outposts?

Per ulteriori informazioni sull'utilizzo dei bucket in S3 su Outposts, consulta Utilizzo di bucket S3 su Outposts.

L'Account AWS che crea il bucket lo possiede ed è l'unico che può eliminarlo.

> **Note**
>
> - Per poter essere eliminati, i bucket Outposts devono essere vuoti.

La console Amazon S3 non supporta operazioni su oggetti di S3 su Outposts. Per eliminare oggetti nel bucket S3 su Outposts, è necessario utilizzare l'API REST, la AWS CLI o gli SDK AWS.

- Prima di poter eliminare un bucket Outposts, è necessario eliminare tutti i punti di accesso Outposts per il bucket. Per ulteriori informazioni, consultare Eliminazione di un punto di accesso.
- Non è possibile recuperare un bucket dopo che è stato eliminato.

Gli esempi seguenti illustrano come eliminare un bucket S3 su Outposts utilizzando la AWS Management Console e AWS Command Line Interface (AWS CLI).

Utilizzo della console S3

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Seleziona il bucket che desideri eliminare e scegli Elimina.
4. Confermare l'eliminazione.

Tramite AWS CLI

L'esempio seguente illustra come eliminare un bucket S3 su Outposts (`s3-outposts:DeleteBucket`) utilizzando la AWS CLI. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
aws s3control delete-bucket --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket
```

Utilizzo dei punti di accesso Amazon S3 su Outposts

Per accedere al bucket Amazon S3 su Outposts devi creare e configurare un punto di accesso.

Gli Access Point semplificano la gestione dell'accesso ai dati su vasta scala per set di dati condivisi in Amazon S3. Gli access point sono endpoint di rete denominati che vengono collegati a bucket che

possono essere utilizzati per eseguire operazioni su oggetti di Amazon S3, ad esempio `GetObject` e `PutObject`. Con S3 su Outposts devi utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outposts. I punti di accesso supportano solo l'indirizzamento in stile hosting virtuale.

> **Note**
>
> L'Account AWS che crea il bucket Outposts lo possiede ed è l'unico che può assegnargli access point.

Nelle sezioni seguenti viene descritto come creare e gestire i punti di accesso per i bucket S3 su Outposts.

Argomenti

Creazione di un punto di accesso S3 su Outposts

Per accedere al bucket Amazon S3 su Outposts devi creare e configurare un punto di accesso.

Gli Access Point semplificano la gestione dell'accesso ai dati su vasta scala per set di dati condivisi in Amazon S3. Gli access point sono endpoint di rete denominati che vengono collegati a bucket che possono essere utilizzati per eseguire operazioni su oggetti di Amazon S3, ad esempio `GetObject` e `PutObject`. Con S3 su Outposts devi utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outposts. I punti di accesso supportano solo l'indirizzamento in stile hosting virtuale.

Gli esempi seguenti illustrano come creare un punto di accesso per S3 su Outposts utilizzando la AWS Management Console, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

> **Note**
>
> L'Account AWS che crea il bucket Outposts lo possiede ed è l'unico che può assegnargli access point.

Utilizzo della console S3

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Seleziona il bucket Outposts per il quale desideri creare un punto di accesso Outposts.
4. Seleziona la scheda Punti di accesso Outposts.
5. Nella sezione Punti di accesso Outposts, seleziona Crea punto di accesso Outposts.
6. Nella sezione Outposts access point settings (Impostazioni punto di accesso Outposts), specifica il nome del punto di accesso e quindi seleziona il cloud privato virtuale (VPC) per il punto di accesso.
7. Se desideri aggiungere una policy per il punto di accesso, inseriscila nella sezione Policy punto di accesso Outposts.

   Per ulteriori informazioni, consulta Configurazione di IAM con S3 su Outposts.

Utilizzo di AWS CLI

Example

Nell'esempio della AWS CLI seguente viene creato un punto di accesso per un bucket di Outposts. Per eseguire questo comando, sostituire *user input placeholders* con le proprie informazioni.

```
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
```

Utilizzo dell'SDK AWS per Java

Example

Nell'esempio SDK per Java seguente viene creato un punto di accesso per un bucket Outposts. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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();

}
```

Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3 su Outposts

Con S3 su Outposts devi utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outposts. Ogni volta che crei un punto di accesso per un bucket, S3 su Outposts genera automaticamente un alias per tale punto di accesso. Puoi utilizzare questo alias del punto di accesso al posto dell'ARN del punto di accesso per qualsiasi operazione del piano dati. Ad esempio, è possibile utilizzare un alias del punto di accesso per eseguire operazioni a livello di oggetto come PUT, GET, LIST e altre ancora. Per un elenco di queste operazioni, consulta Operazioni API Amazon S3 per la gestione degli oggetti.

Negli esempi seguenti viene illustrato un ARN e un alias per un punto di accesso denominato *my-access-point*.

- ARN del punto di accesso - `arn:aws:s3-outposts:`*region*`:`*123456789012*`:outpost/`*op-01ac5d28a6a232904*`/accesspoint/`*my-access-point*

- Alias del punto di accesso - *my-access-po-o01ac5d28a6a232904e8xz5w8ijx1qzlbp3i3kuse10*--op-s3

Per ulteriori informazioni sull'utilizzo degli ARN, consulta Amazon Resource Name (ARN) nella Riferimenti generali di AWS.

Per ulteriori informazioni sugli alias del punto di accesso, consulta gli argomenti indicati di seguito.

Argomenti

- Alias del punto di accesso
- Utilizzo di un alias del punto di accesso in un'operazione di oggetto S3 su Outposts
- Restrizioni

Alias del punto di accesso

Un alias del punto di accesso viene creato nello stesso spazio dei nomi del bucket S3 su Outposts. Quando crei un punto di accesso, S3 su Outposts genera automaticamente un alias per tale punto di accesso che non può essere modificato. Un alias del punto di accesso soddisfa tutti i requisiti di un nome di bucket S3 su Outposts valido e comprende le seguenti parti:

access point name prefix-metadata--op-s3

> **Note**
>
> Il suffisso `--op-s3` è riservato agli alias del punto di accesso; ti consigliamo di non utilizzarlo per i nomi dei bucket o dei punti di accesso. Per ulteriori informazioni sulle regole di denominazione dei bucket S3 su Outposts, consulta Utilizzo di bucket S3 su Outposts.

Ricerca dell'alias del punto di accesso

Negli esempi seguenti viene illustrato come trovare un alias del punto di accesso utilizzando la console Amazon S3 e AWS CLI.

Example - Ricerca e copia dell'alias del punto di accesso nella console Amazon S3

Dopo aver creato un punto di accesso nella console, puoi recuperarne l'alias nella colonna Access Point alias (Alias del punto di accesso) nell'elenco Access Points (Punti di accesso).

Copia dell'alias del punto di accesso

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts access points (Punti di accesso Outposts).
3. Per copiare l'alias del punto di accesso, effettua una delle seguenti operazioni:
   - Nell'elenco Access Points (Punti di accesso), seleziona il pulsante di opzione accanto al nome del punto di accesso, quindi scegli Copy Access Point alias (Copia alias del punto di accesso).
   - Scegliere il nome del punto di accesso. Quindi, in Outposts access point overview (Panoramica dei punti di accesso Outposts), copia l'alias del punto di accesso.

Example - Creazione di un punto di accesso utilizzando la AWS CLI e ricerca dell'alias del punto di accesso nella risposta

Il seguente esempio del comando `create-access-point` nella AWS CLI crea il punto di accesso e restituisce l'alias del punto di accesso generato automaticamente. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
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 - Recupero di un alias del punto di accesso utilizzando la AWS CLI

L'esempio AWS CLI seguente del comando `get-access-point` restituisce le informazioni sul punto di accesso specificato. Queste informazioni includono l'alias del punto di accesso. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
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 - Elenco dei punti di accesso per trovare un alias del punto di accesso utilizzando la AWS CLI

Il seguente esempio del comando `list-access-points` nella AWS CLI elenca le informazioni sul punto di accesso specificato. Queste informazioni includono l'alias del punto di accesso. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
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"
        }
    ]
```

```
}
```

Utilizzo di un alias del punto di accesso in un'operazione di oggetto S3 su Outposts

Quando si adottano i punti di accesso, è possibile utilizzare l'alias del punto di accesso senza richiedere importanti modifiche di codice.

Questo esempio nella AWS CLI mostra un'operazione `get-object` per un bucket S3 su Outposts. Questo esempio utilizza l'alias del punto di accesso come valore per `--bucket` anziché l'ARN completo del punto di accesso.

```
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": {}
}
```

Restrizioni

- Gli alias non possono essere configurati dai clienti.
- Gli alias non possono essere eliminati, modificati o disabilitati in un punto di accesso.
- Non puoi utilizzare un alias del punto di accesso per le operazioni del piano di controllo (control-plane) S3 su Outposts. Per l'elenco delle operazioni del piano di controllo (control-plane) S3 su Outposts, consulta Operazioni API Amazon S3 Control per la gestione dei bucket.
- Gli alias non possono essere utilizzati nelle policy AWS Identity and Access Management (IAM).

Visualizzazione delle informazioni sulla configurazione di un punto di accesso

Gli Access Point semplificano la gestione dell'accesso ai dati su vasta scala per set di dati condivisi in Amazon S3. Gli access point sono endpoint di rete denominati che vengono collegati a bucket che possono essere utilizzati per eseguire operazioni su oggetti di Amazon S3, ad esempio `GetObject` e

`PutObject`. Con S3 su Outposts devi utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outposts. I punti di accesso supportano solo l'indirizzamento in stile hosting virtuale.

Gli argomenti seguenti descrivono come ottenere le informazioni della configurazione di un punto di accesso Amazon S3 su Outposts utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts access points (Punti di accesso Outposts).
3. Scegli il punto di accesso Outposts per cui desideri visualizzare i dettagli di configurazione.
4. In Outposts access point overview (Panoramica del punto di accesso Outposts), esamina i dettagli della configurazione del punto di accesso.

Tramite AWS CLI

Nell'esempio della AWS CLI seguente si ottiene un punto di accesso per un bucket di Outposts. Sostituire *`user input placeholders`* con le proprie informazioni.

```
aws s3control get-access-point --account-id 123456789012 --name arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

Utilizzo dell'SDK AWS per Java

Nell'esempio SDK per Java seguente si ottiene un punto di accesso per un bucket 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());
```

```
}
```

Visualizzazione dell'elenco dei punti di accesso Amazon S3 su Outposts

Gli Access Point semplificano la gestione dell'accesso ai dati su vasta scala per set di dati condivisi in Amazon S3. Gli access point sono endpoint di rete denominati che vengono collegati a bucket che possono essere utilizzati per eseguire operazioni su oggetti di Amazon S3, ad esempio `GetObject` e `PutObject`. Con S3 su Outposts devi utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outposts. I punti di accesso supportano solo l'indirizzamento in stile hosting virtuale.

Gli argomenti seguenti descrivono come ottenere l'elenco dei punti di accesso Amazon S3 su Outposts utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts access points (Punti di accesso Outposts).
3. In Outposts access points (Punti di accesso Outposts), consulta l'elenco dei punti di accesso S3 su Outposts.

Tramite AWS CLI

Nell'esempio della AWS CLI seguente vengono elencati i punti di accesso per un bucket di Outposts. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
aws s3control list-access-points --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket
```

Utilizzo dell'SDK AWS per Java

Nell'esempio SDK per Java seguente vengono elencati i punti di accesso per un bucket 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());

}
```

Eliminazione di un punto di accesso

Gli Access Point semplificano la gestione dell'accesso ai dati su vasta scala per set di dati condivisi in Amazon S3. Gli access point sono endpoint di rete denominati che vengono collegati a bucket che possono essere utilizzati per eseguire operazioni su oggetti di Amazon S3, ad esempio `GetObject` e `PutObject`. Con S3 su Outposts devi utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outposts. I punti di accesso supportano solo l'indirizzamento in stile hosting virtuale.

Gli esempi seguenti illustrano come eliminare un punto di accesso utilizzando la AWS Management Console e la AWS Command Line Interface (AWS CLI).

Utilizzo della console S3

1. Apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts access points (Punti di accesso Outposts).
3. Nella sezione Punti di accesso Outposts, seleziona il punto di accesso Outposts da eliminare.
4. Scegliere Delete (Elimina).
5. Confermare l'eliminazione.

Tramite AWS CLI

Il seguente esempio di AWS CLI mostra come eliminare un punto di accesso su Outposts. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
aws s3control delete-access-point --account-id 123456789012 --name arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

Aggiunta o modifica di una policy del punto di accesso

Ogni punto di accesso dispone di autorizzazioni e controlli di rete distinti che Amazon S3 su Outposts applica per qualsiasi richiesta effettuata tramite il punto di accesso. Ogni access point applica una policy di access point personalizzata che funziona in combinazione con la policy di bucket collegata al bucket sottostante. Per ulteriori informazioni, consultare Access point.

Nei seguenti argomenti viene descritto come aggiungere o modificare la policy del punto di accesso per Amazon S3 su Outposts utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Seleziona il bucket Outposts per il quale desideri modificare la policy del punto di accesso.
4. Seleziona la scheda Punti di accesso Outposts.
5. Nella sezione Punti di accesso Outposts, seleziona il punto di accesso di cui desideri modificare la policy e scegli Modifica policy.
6. Aggiungi o modifica la policy nella sezione Policy punto di accesso Outposts . Per ulteriori informazioni, consulta Configurazione di IAM con S3 su Outposts.

Utilizzo di AWS CLI

Nell'esempio della AWS CLI seguente viene inserita una policy su un punto di accesso Outposts.

1. Salvare la seguente policy del punto di accesso in un file JSON. In questo esempio, il file è denominato `appolicy1.json`. Sostituire *`user input placeholders`* con le proprie informazioni.

```
{
   "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. Inviare il file JSON come parte del comando CLI `put-access-point-policy`. Sostituire *`user input placeholders`* con le proprie informazioni.

```
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
```

Utilizzo dell'SDK AWS per Java

Nell'esempio SDK per Java seguente viene inserita una policy su un punto di accesso 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());
```

```
}
```

Visualizzazione della policy per un punto di accesso S3 su Outposts

Ogni punto di accesso dispone di autorizzazioni e controlli di rete distinti che Amazon S3 su Outposts applica a qualsiasi richiesta effettuata tramite il punto di accesso. Ogni access point applica una policy di access point personalizzata che funziona in combinazione con la policy di bucket collegata al bucket sottostante. Per ulteriori informazioni, consulta Access point.

Per ulteriori informazioni sull'utilizzo dei punti di accesso in S3 su Outposts, consulta Utilizzo di bucket S3 su Outposts.

Nei seguenti argomenti viene descritto come visualizzare la policy del punto di accesso di Amazon S3 su Outposts utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts access points (Punti di accesso Outposts).
3. Seleziona il punto di accesso Outposts per il quale desideri visualizzare la policy.
4. Nella scheda Permissions (Autorizzazioni) esamina la policy del punto di accesso S3 su Outposts.
5. Per modificare la policy del punto di accesso, consulta Aggiunta o modifica di una policy del punto di accesso.

Tramite AWS CLI

Nell'esempio della AWS CLI seguente viene ottenuta una policy per un punto di accesso Outposts. Per eseguire questo comando, sostituire *`user input placeholders`* con le proprie informazioni.

```
aws s3control get-access-point-policy --account-id 123456789012 --name arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

Utilizzo dell'SDK AWS per Java

Nell'esempio SDK per Java seguente si ottiene una policy per un punto di accesso di 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());

}
```

Utilizzo degli endpoint Amazon S3 su Outposts

Per instradare le richieste verso un punto di accesso Amazon S3 su Outposts, è necessario creare e configurare un endpoint S3 su Outposts. Per creare un endpoint, è necessario disporre di una connessione attiva con il collegamento del servizio alla regione di origine degli outpost. Ogni cloud privato virtuale (VPC) del tuo Outpost può avere un endpoint associato. Per ulteriori informazioni sull'endpoint, consulta Requisiti di rete di S3 su Outposts. È necessario creare un endpoint per poter accedere ai bucket Outposts ed eseguire operazioni sugli oggetti. Per ulteriori informazioni, consulta Endpoint.

Dopo aver creato un endpoint, puoi utilizzare il campo 'Stato' per informazioni sullo stato corrente dell'endpoint. Se gli outpost sono offline, verrà restituito il valore CREATE_FAILED. È possibile verificare la connessione del collegamento del servizio, eliminare l'endpoint e riprovare l'operazione di creazione dopo avere ristabilito la connessione. Per un elenco dei codici di errore aggiuntivi, vedi più avanti. Per ulteriori informazioni, consulta Endpoint.

API	Stato	Codice di errore del motivo dell'operazione non riuscita	Messaggio - Motivo dell'operazione non riuscita
`CreateEndpoint`	Create_Failed	OutpostNotReachable	Impossibile creare l'endpoint perché la connessione del collegamento del

API	Stato	Codice di errore del motivo dell'operazione non riuscita	Messaggio - Motivo dell'operazione non riuscita
			servizio alla regione principale degli outpost è inattiva. Controlla la connessio ne, elimina l'endpoint e riprova.
`CreateEndpoint`	Create_Failed	InternalError	Impossibile creare l'endpoint a causa di un errore interno. Elimina l'endpoint e crealo di nuovo.
`DeleteEndpoint`	Delete_Failed	OutpostNotReachable	Impossibile eliminare l'endpoint perché la connessione del collegamento del servizio alla regione principale degli outpost è inattiva. Controlla la connessio ne e riprova.
`DeleteEndpoint`	Delete_Failed	InternalError	Impossibile eliminare l'endpoint a causa di un errore interno. Riprova.

Per ulteriori informazioni sull'utilizzo dei bucket in S3 su Outposts, consulta Utilizzo di bucket S3 su Outposts.

Le seguenti sezioni descrivono come creare e gestire gli endpoint per S3 su Outposts.

Argomenti

- Creazione di un endpoint in un Outpost
- Visualizzazione dell'elenco degli endpoint Amazon S3 su Outposts
- Eliminazione di un endpoint Amazon S3 su Outposts

Creazione di un endpoint in un Outpost

Per instradare le richieste verso un punto di accesso Amazon S3 su Outposts, è necessario creare e configurare un endpoint S3 su Outposts. Per creare un endpoint, è necessario disporre di una connessione attiva con il collegamento del servizio alla regione di origine degli outpost. Ogni cloud privato virtuale (VPC) del tuo Outpost può avere un endpoint associato. Per ulteriori informazioni

sull'endpoint, consulta Requisiti di rete di S3 su Outposts. È necessario creare un endpoint per poter accedere ai bucket Outposts ed eseguire operazioni sugli oggetti. Per ulteriori informazioni, consulta Endpoint.

Autorizzazioni

Per ulteriori informazioni sulle autorizzazioni richieste per la creazione di un endpoint, consulta Autorizzazioni per endpoint S3 su Outposts.

Quando crei un endpoint, S3 Outposts crea anche un ruolo collegato al servizio nel tuo Account AWS. Per ulteriori informazioni, consulta Utilizzo dei ruoli collegati ai servizi per Amazon S3 su Outposts.

Gli esempi seguenti illustrano come creare un endpoint S3 su Outposts utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts access points (Punti di accesso Outposts).
3. Seleziona la scheda Outposts endpoints (Endpoint Outposts).
4. Scegli Create Outposts endpoint (Crea endpoint Outposts).
5. In Outpost, scegli l'Outpost su cui creare questo endpoint.
6. In VPC, scegli un VPC che non disponga ancora di un endpoint e rispetti le regole degli endpoint di Outposts.

   Un Virtual Private Cloud (VPC) consente di avviare risorse AWS in una rete virtuale definita dall'utente. Questa rete virtuale è simile a una comune rete da gestire all'interno del proprio data center, ma con i vantaggi dell'impiego dell'infrastruttura scalabile di AWS.

   Se non si dispone di un VPC, scegliere Crea VPC. Per ulteriori informazioni, consulta Creazione di access point limitati a un cloud privato virtuale.
7. Scegli Create Outposts endpoint (Crea endpoint Outposts).

Utilizzo di AWS CLI

Example

Nel seguente esempio AWS CLI viene creato un endpoint per un Outpost utilizzando il tipo di accesso alle risorse del VPC. Il VPC deriva dalla sottorete. Per eseguire questo comando, sostituire *user input placeholders* con le proprie informazioni.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
 subnet-8c7a57c5 --security-group-id sg-ab19e0d1
```

Nel seguente esempio AWS CLI viene creato un endpoint per un Outpost utilizzando il tipo di accesso al pool di indirizzi IP di proprietà del cliente (pool CoIP). Per eseguire questo comando, sostituire *user input placeholders* con le proprie informazioni.

```
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
```

Utilizzo dell'SDK AWS per Java

Example

Nell'esempio SDK per Java seguente viene creato un endpoint per un outpost. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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());
}
```

Visualizzazione dell'elenco degli endpoint Amazon S3 su Outposts

Per instradare le richieste verso un punto di accesso Amazon S3 su Outposts, è necessario creare e configurare un endpoint S3 su Outposts. Per creare un endpoint, è necessario disporre di una connessione attiva con il collegamento del servizio alla regione di origine degli outpost. Ogni cloud privato virtuale (VPC) del tuo Outpost può avere un endpoint associato. Per ulteriori informazioni sull'endpoint, consulta Requisiti di rete di S3 su Outposts. È necessario creare un endpoint per poter accedere ai bucket Outposts ed eseguire operazioni sugli oggetti. Per ulteriori informazioni, consulta Endpoint.

Gli esempi seguenti mostrano come ottenere l'elenco degli endpoint S3 su Outposts utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts access points (Punti di accesso Outposts).
3. Nella pagina Outposts access points (Punti di accesso Outposts) seleziona la scheda Outposts endpoints (Endpoint Outposts).
4. In Outposts endpoints (Endpoint Outposts), puoi visualizzare un elenco dei tuoi endpoint S3 su Outposts.

Utilizzo di AWS CLI

Nel seguente esempio AWS CLI sono riportati gli endpoint per le risorse AWS Outposts associate all'account. Per ulteriori informazioni su questo comando, consulta list-endpoints nellaGuida di riferimento a AWS CLI.

```
aws s3outposts list-endpoints
```

Utilizzo dell'SDK AWS per Java

Nell'esempio SDK per Java seguente vengono elencati gli endpoint per un outpost. Per ulteriori informazioni, consulta ListEndpoints nella Documentazione di riferimento delle API di 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);
}
```

Eliminazione di un endpoint Amazon S3 su Outposts

Per instradare le richieste verso un punto di accesso Amazon S3 su Outposts, è necessario creare e configurare un endpoint S3 su Outposts. Per creare un endpoint, è necessario disporre di una connessione attiva con il collegamento del servizio alla regione di origine degli outpost. Ogni cloud privato virtuale (VPC) del tuo Outpost può avere un endpoint associato. Per ulteriori informazioni sull'endpoint, consulta Requisiti di rete di S3 su Outposts. È necessario creare un endpoint per poter accedere ai bucket Outposts ed eseguire operazioni sugli oggetti. Per ulteriori informazioni, consulta Endpoint.

Gli esempi seguenti mostrano come eliminare gli endpoint S3 su Outposts utilizzando la AWS Management Console, AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Utilizzo della console S3

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts access points (Punti di accesso Outposts).
3. Nella pagina Outposts access points (Punti di accesso Outposts) seleziona la scheda Outposts endpoints (Endpoint Outposts).

4. In Outposts endpoints (Endpoint Outposts) scegli l'endpoint che desideri eliminare e seleziona Delete (Elimina).

Utilizzo di AWS CLI

Nell'esempio della AWS CLI seguente viene eliminato un endpoint per un Outpost. Per eseguire questo comando, sostituire *user input placeholders* con le proprie informazioni.

```
aws s3outposts delete-endpoint --endpoint-id example-endpoint-id --outpost-
id op-01ac5d28a6a232904
```

Utilizzo dell'SDK AWS per Java

Nell'esempio SDK per Java seguente viene eliminato un endpoint per un Outpost. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
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.");
}
```

Utilizzo di oggetti S3 su Outposts

Con Amazon S3 on Outposts, puoi creare bucket S3 sui tuoi AWS Outposts e archiviare e recuperare facilmente oggetti in locale per applicazioni che richiedono l'accesso locale ai dati, l'elaborazione

locale dei dati e la residenza dei dati. S3 on Outposts offre una nuova classe di storage, S3 Outposts OUTPOSTS (), che utilizza le API di Amazon S3 ed è progettata per archiviare i dati in modo duraturo e ridondante su più dispositivi e server sul tuo. AWS Outposts Comunichi con il bucket Outpost utilizzando un punto di accesso e una connessione di endpoint su un Virtual Private Cloud (VPC). Sui bucket Outposts puoi utilizzare le stesse API e caratteristiche di Amazon S3, comprese policy di accesso, crittografia e tagging. Puoi usare S3 su Outposts tramite AWS Management Console AWS Command Line Interface ,AWS CLI() AWS , SDK o API REST.

Gli oggetti sono le entità fondamentali archiviate in Amazon S3 su Outposts. Ogni oggetto è contenuto in un bucket. È necessario utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outpost. Quando specifichi il bucket per le operazioni di oggetto, utilizza il nome della risorsa Amazon (ARN) o l'alias del punto di accesso. Per ulteriori informazioni sugli alias del punto di accesso, consulta Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3 su Outposts.

L'esempio seguente mostra il formato ARN per S3 sui punti di accesso Outposts, che include il Regione AWS codice per la regione in cui risiede l'Outpost, l'ID, l' Account AWS ID Outpost e il nome del punto di accesso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Per ulteriori informazioni sugli ARN S3 su Outposts, consulta ARN delle risorse per S3 su Outposts.

Gli ARN degli oggetti utilizzano il seguente formato, che include il nome a cui appartiene l' Regione AWS Outpost, l'ID, l' Account AWS ID Outpost, il nome del bucket e la chiave dell'oggetto:

```
arn:aws:s3-outposts:us-west-2:123456789012:outpost/ op-01ac5d28a6a232904/
bucket/example-s3-bucket1/object/myobject
```

Con Amazon S3 su Outposts, i dati degli oggetti vengono sempre archiviati nell'Outpost. Quando si AWS installa un rack Outpost, i dati rimangono locali rispetto a Outpost per soddisfare i requisiti di residenza dei dati. I tuoi oggetti non lasciano mai il tuo Outpost e non sono in una Regione AWS. Poiché AWS Management Console è ospitato in una regione, non puoi utilizzare la console per caricare o gestire oggetti in Outpost. Tuttavia, puoi utilizzare l'API REST, AWS Command Line Interface (AWS CLI) e gli AWS SDK per caricare e gestire gli oggetti tramite i tuoi punti di accesso.

Argomenti

- Carica un oggetto in un bucket S3 on Outposts
- Copia di un oggetto in un bucket Amazon S3 su Outposts utilizzando AWS SDK for Java

- Recupero di un oggetto da un bucket Amazon S3 su Outposts
- Elenco di oggetti in un bucket Amazon S3 su Outposts
- Eliminazione di oggetti nei bucket Amazon S3 su Outposts
- Utilizzo di HeadBucket per determinare se esiste un bucket S3 su Outposts e sono disponibili le autorizzazioni di accesso
- Esecuzione e gestione di un caricamento in più parti con SDK per Java
- Utilizzo di URL prefirmati per S3 su Outposts
- Amazon S3 su Outposts con Amazon EMR locale su Outposts
- Memorizzazione nella cache di autorizzazione e autenticazione

Carica un oggetto in un bucket S3 on Outposts

Gli oggetti sono le entità fondamentali archiviate in Amazon S3 su Outposts. Ogni oggetto è contenuto in un bucket. È necessario utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outpost. Quando specifichi il bucket per le operazioni di oggetto, utilizza il nome della risorsa Amazon (ARN) o l'alias del punto di accesso. Per ulteriori informazioni sugli alias del punto di accesso, consulta Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3 su Outposts.

L'esempio seguente mostra il formato ARN per S3 sui punti di accesso Outposts, che include il Regione AWS codice per la regione in cui risiede l'Outpost, l'ID, l' Account AWS ID Outpost e il nome del punto di accesso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Per ulteriori informazioni sugli ARN S3 su Outposts, consulta ARN delle risorse per S3 su Outposts.

Con Amazon S3 su Outposts, i dati degli oggetti vengono sempre archiviati nell'Outpost. Quando si AWS installa un rack Outpost, i dati rimangono locali rispetto a Outpost per soddisfare i requisiti di residenza dei dati. I tuoi oggetti non lasciano mai il tuo Outpost e non sono in una Regione AWS. Poiché AWS Management Console è ospitato in una regione, non puoi utilizzare la console per caricare o gestire oggetti in Outpost. Tuttavia, puoi utilizzare l'API REST, AWS Command Line Interface (AWS CLI) e gli AWS SDK per caricare e gestire gli oggetti tramite i tuoi punti di accesso.

I seguenti AWS CLI AWS SDK for Java esempi mostrano come caricare un oggetto su un bucket S3 on Outposts utilizzando un punto di accesso.

AWS CLI

Example

Nell'esempio seguente viene inserito un oggetto denominato `sample-object.xml` in un bucket S3 su Outposts (`s3-outposts:PutObject`) utilizzando AWS CLI. Per usare questo comando, sostituire *`user input placeholder`* con le proprie informazioni. Per ulteriori informazioni su questo comando, consulta put-object nella Guida di riferimento di 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

Nell'esempio seguente viene inserito un oggetto in un bucket S3 su Outposts utilizzando SDK per Java. Per utilizzare questo esempio, sostituire *`user input placeholder`* con le tue informazioni. Per ulteriori informazioni, consulta Caricamento degli oggetti.

```
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 di un oggetto in un bucket Amazon S3 su Outposts utilizzando AWS SDK for Java

Gli oggetti sono le entità fondamentali archiviate in Amazon S3 su Outposts. Ogni oggetto è contenuto in un bucket. È necessario utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outpost. Quando specifichi il bucket per le operazioni di oggetto, utilizza il nome della risorsa Amazon (ARN) o l'alias del punto di accesso. Per ulteriori informazioni sugli alias del punto di accesso, consulta Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3 su Outposts.

Nell'esempio seguente viene illustrato il formato ARN per i punti di accesso S3 su Outposts, che include il codice Regione AWS per la Regione in cui si trova l'Outpost, l'ID Account AWS, l'ID Outpost e il nome del punto di accesso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Per ulteriori informazioni sugli ARN S3 su Outposts, consulta ARN delle risorse per S3 su Outposts.

Con Amazon S3 su Outposts, i dati degli oggetti vengono sempre archiviati nell'Outpost. Quando AWS installa un rack Outpost, i tuoi dati rimangono locali nel tuo Outpost per soddisfare i requisiti di residenza dei dati. I tuoi oggetti non lasciano mai il tuo Outpost e non sono in una Regione AWS. Dal momento che la AWS Management Console è ospitata nella regione, non puoi utilizzare la console per caricare o gestire oggetti nel tuo Outpost. Tuttavia, puoi utilizzare l'API REST, AWS Command Line Interface (AWS CLI) e gli SDK AWS per caricare e gestire gli oggetti tramite i punti di accesso.

L'esempio seguente illustra come copiare un oggetto in un bucket S3 su Outposts utilizzando la AWS SDK for Java.

Utilizzo dell'SDK AWS per Java

Nell'esempio S3 su Outposts seguente un oggetto viene copiato in un nuovo oggetto nello stesso bucket utilizzando l'SDK per Java. Per utilizzare questo comando, sostituisci *`user input placeholders`* con le tue informazioni.

```
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();
        }
    }
}
```

Recupero di un oggetto da un bucket Amazon S3 su Outposts

Gli oggetti sono le entità fondamentali archiviate in Amazon S3 su Outposts. Ogni oggetto è contenuto in un bucket. È necessario utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outpost. Quando specifichi il bucket per le operazioni di oggetto, utilizza il nome della risorsa Amazon (ARN) o l'alias del punto di accesso. Per ulteriori informazioni sugli alias del punto di accesso, consulta Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3 su Outposts.

Nell'esempio seguente viene illustrato il formato ARN per i punti di accesso S3 su Outposts, che include il codice Regione AWS per la Regione in cui si trova l'Outpost, l'ID Account AWS, l'ID Outpost e il nome del punto di accesso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Per ulteriori informazioni sugli ARN S3 su Outposts, consulta ARN delle risorse per S3 su Outposts.

Con Amazon S3 su Outposts, i dati degli oggetti vengono sempre archiviati nell'Outpost. Quando AWS installa un rack Outpost, i tuoi dati rimangono locali nel tuo Outpost per soddisfare i requisiti di residenza dei dati. I tuoi oggetti non lasciano mai il tuo Outpost e non sono in una Regione AWS. Dal momento che la AWS Management Console è ospitata nella regione, non puoi utilizzare la console per caricare o gestire oggetti nel tuo Outpost. Tuttavia, puoi utilizzare l'API REST, AWS Command Line Interface (AWS CLI) e gli SDK AWS per caricare e gestire gli oggetti tramite i punti di accesso.

Gli esempi seguenti illustrano come scaricare un oggetto utilizzando AWS Command Line Interface (AWS CLI) eAWS SDK for Java.

Tramite AWS CLI

Nell'esempio seguente viene inserito un oggetto denominato `sample-object.xml` da un bucket S3 su Outposts (`s3-outposts:GetObject`) utilizzando AWS CLI. Per usare questo comando, sostituire ogni *`user input placeholder`* con le proprie informazioni. Per ulteriori informazioni su questo comando, consulta get-object nella Guida di riferimento a 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
```

Utilizzo dell'SDK AWS per Java

Nell'esempio S3 su Outposts seguente viene ottenuto un oggetto utilizzando SDK per Java. Per utilizzare questo comando, sostituisci *`user input placeholder`* con le tue informazioni. Per ulteriori informazioni, consulta GetObject nella Documentazione di riferimento delle API di 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();
    }
}
```

Elenco di oggetti in un bucket Amazon S3 su Outposts

Gli oggetti sono le entità fondamentali archiviate in Amazon S3 su Outposts. Ogni oggetto è contenuto in un bucket. È necessario utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outpost. Quando specifichi il bucket per le operazioni di oggetto, utilizza il nome della risorsa Amazon (ARN) o l'alias del punto di accesso. Per ulteriori informazioni sugli alias del punto di accesso, consulta Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3 su Outposts.

Nell'esempio seguente viene illustrato il formato ARN per i punti di accesso S3 su Outposts, che include il codice Regione AWS per la Regione in cui si trova l'Outpost, l'ID Account AWS, l'ID Outpost e il nome del punto di accesso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Per ulteriori informazioni sugli ARN S3 su Outposts, consulta ARN delle risorse per S3 su Outposts.

> **Note**
>
> Con Amazon S3 su Outposts, i dati degli oggetti vengono sempre archiviati nell'Outpost. Quando AWS installa un rack Outpost, i tuoi dati rimangono locali nel tuo Outpost per soddisfare i requisiti di residenza dei dati. I tuoi oggetti non lasciano mai il tuo Outpost e non

sono in una Regione AWS. Dal momento che la AWS Management Console è ospitata nella regione, non puoi utilizzare la console per caricare o gestire oggetti nel tuo Outpost. Tuttavia, puoi utilizzare l'API REST, AWS Command Line Interface (AWS CLI) e gli SDK AWS per caricare e gestire gli oggetti tramite i punti di accesso.

Gli esempi seguenti illustrano come elencare gli oggetti in un bucket S3 su Outposts utilizzando AWS CLI e AWS SDK for Java.

Tramite AWS CLI

Nell'esempio seguente sono riportati gli oggetti in un bucket S3 su Outposts (`s3-outposts:ListObjectsV2`) tramite AWS CLI. Per usare questo comando, sostituire *`user input placeholder`* con le proprie informazioni. Per ulteriori informazioni su questo comando, consulta list-object-v2 nella Guida di riferimento di AWS CLI.

```
aws s3api list-objects-v2 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

Note

Utilizzando questa operazione con Amazon S3 su Outposts tramite le AWS SDK, fornisci l'ARN del punto di accesso Outposts invece del nome del bucket nella seguente scheda: `arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-Outposts-Access-Point`. Per ulteriori informazioni sugli ARN S3 su Outposts, consulta ARN delle risorse per S3 su Outposts.

Utilizzo dell'SDK AWS per Java

Nell'esempio S3 su Outposts seguente vengono elencati oggetti in un bucket utilizzando SDK per Java. Per usare questo comando, sostituire *`user input placeholder`* con le proprie informazioni.

Important

Questo esempio usa ListObjectsV2, che è l'ultima revisione dell'operazione API `ListObjects`. Si consiglia di utilizzare questa operazione API rivista per lo sviluppo di

applicazioni. Per la compatibilità con le versioni precedenti, Amazon S3 continua a supportare la versione precedente di questa operazione 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();
        }
    }
}
```

Eliminazione di oggetti nei bucket Amazon S3 su Outposts

Gli oggetti sono le entità fondamentali archiviate in Amazon S3 su Outposts. Ogni oggetto è contenuto in un bucket. È necessario utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outpost. Quando specifichi il bucket per le operazioni di oggetto, utilizza il nome della risorsa Amazon (ARN) o l'alias del punto di accesso. Per ulteriori informazioni sugli alias del punto di accesso, consulta Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3 su Outposts.

Nell'esempio seguente viene illustrato il formato ARN per i punti di accesso S3 su Outposts, che include il codice Regione AWS per la Regione in cui si trova l'Outpost, l'ID Account AWS, l'ID Outpost e il nome del punto di accesso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Per ulteriori informazioni sugli ARN S3 su Outposts, consulta ARN delle risorse per S3 su Outposts.

Con Amazon S3 su Outposts, i dati degli oggetti vengono sempre archiviati nell'Outpost. Quando AWS installa un rack Outpost, i tuoi dati rimangono locali nel tuo Outpost per soddisfare i requisiti di residenza dei dati. I tuoi oggetti non lasciano mai il tuo Outpost e non sono in una Regione AWS. Dal momento che la AWS Management Console è ospitata nella regione, non puoi utilizzare la console per caricare o gestire oggetti nel tuo Outpost. Tuttavia, puoi utilizzare l'API REST, AWS Command Line Interface (AWS CLI) e gli SDK AWS per caricare e gestire gli oggetti tramite i punti di accesso.

Negli esempi seguenti viene illustrato come eliminare un singolo oggetto o più oggetti in un bucket Amazon S3 su Outposts utilizzando AWS Command Line Interface (AWS CLI) e AWS SDK for Java.

Tramite AWS CLI

Negli esempi seguenti viene illustrato come eliminare un singolo oggetto o più oggetti in un bucket S3 su Outposts.

delete-object

Nell'esempio seguente viene eliminato un oggetto denominato `sample-object.xml` da un bucket S3 su Outposts (`s3-outposts:DeleteObject`) utilizzando AWS CLI. Per eseguire questo comando, sostituire *`user input placeholder`* con le proprie informazioni. Per ulteriori informazioni su questo comando, consulta delete-object nella Guida di riferimento 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

Nell'esempio seguente viene eliminato un oggetto denominato `sample-object.xml` e `test1.text` da un bucket S3 su Outposts (`s3-outposts:DeleteObject`) utilizzando AWS CLI. Per eseguire questo comando, sostituire *`user input placeholder`* con le proprie informazioni. Per ulteriori informazioni su questo comando, consulta delete-objects nella Guida di riferimento 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
```

```
}
```

Utilizzo dell'SDK AWS per Java

Negli esempi seguenti viene illustrato come eliminare un singolo oggetto o più oggetti in un bucket S3 su Outposts.

DeleteObject

Nell'esempio S3 su Outposts seguente viene eliminato un oggetto in un bucket utilizzando SDK per Java. Per utilizzare questo esempio, specificare il punto di accesso ARN per l'Outpost e il nome della chiave dell'oggetto che si desidera eliminare. Per ulteriori informazioni, consulta DeleteObject nella Documentazione di riferimento delle API di 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

Nell'esempio S3 su Outposts seguente sono caricati e poi eliminati oggetti in un bucket utilizzando SDK per Java. Per utilizzare questo esempio, specificare il punto di accesso ARN per l'Outpost. Per ulteriori informazioni, consulta DeleteObjects nella Documentazione di riferimento delle API di 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();
        }
    }
}
```

Utilizzo di HeadBucket per determinare se esiste un bucket S3 su Outposts e sono disponibili le autorizzazioni di accesso

Gli oggetti sono le entità fondamentali archiviate in Amazon S3 su Outposts. Ogni oggetto è contenuto in un bucket. È necessario utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outpost. Quando specifichi il bucket per le operazioni di oggetto, utilizza il nome della risorsa Amazon (ARN) o l'alias del punto di accesso. Per ulteriori informazioni sugli alias del punto di accesso, consulta Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3 su Outposts.

Nell'esempio seguente viene illustrato il formato ARN per i punti di accesso S3 su Outposts, che include il codice Regione AWS per la Regione in cui si trova l'Outpost, l'ID Account AWS, l'ID Outpost e il nome del punto di accesso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Per ulteriori informazioni sugli ARN S3 su Outposts, consulta ARN delle risorse per S3 su Outposts.

> **Note**
>
> Con Amazon S3 su Outposts, i dati degli oggetti vengono sempre archiviati nell'Outpost. Quando AWS installa un rack Outpost, i tuoi dati rimangono locali nel tuo Outpost per soddisfare i requisiti di residenza dei dati. I tuoi oggetti non lasciano mai il tuo Outpost e non sono in una Regione AWS. Dal momento che la AWS Management Console è ospitata nella regione, non puoi utilizzare la console per caricare o gestire oggetti nel tuo Outpost. Tuttavia, puoi utilizzare l'API REST, AWS Command Line Interface (AWS CLI) e gli SDK AWS per caricare e gestire gli oggetti tramite i punti di accesso.

I seguenti esempi AWS Command Line Interface (AWS CLI) e AWS SDK for Java illustrano come utilizzare l'operazione API HeadBucket per determinare se esiste un bucket Amazon S3 su Outposts e se sono disponibili le autorizzazioni per accedervi. Per ulteriori informazioni, consulta HeadBucket nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Tramite AWS CLI

Nell'esempio AWS CLI di S3 su Outposts seguente viene utilizzato il comando `head-bucket` per determinare se esiste un bucket e sono disponibili le autorizzazioni per accedervi. Per usare questo comando, sostituire *`user input placeholder`* con le proprie informazioni. Per ulteriori informazioni su questo comando, consulta head-bucket nella Guida di riferimento di AWS CLI.

```
aws s3api head-bucket --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

Utilizzo dell'SDK AWS per Java

Nell'esempio S3 su Outposts seguente viene illustrato come determinare se esiste un bucket e sono disponibili le autorizzazioni per accedervi. Per utilizzare questo esempio, specificare il punto di accesso ARN per l'Outpost. Per ulteriori informazioni, consulta HeadBucket nella Documentazione di riferimento delle API di 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();
        }
    }
}
```

Esecuzione e gestione di un caricamento in più parti con SDK per Java

Con Amazon S3 su Outposts è possibile creare bucket S3 sulle risorse AWS Outposts, nonché archiviare e recuperare gli oggetti in locale per le applicazioni che richiedono l'accesso ai dati in locale, l'elaborazione dei dati in locale e la residenza dei dati. Puoi utilizzare S3 su Outposts tramite la AWS Management Console, AWS Command Line Interface (AWS CLI), gli SDK AWS o l'API REST. Per ulteriori informazioni, consulta Che cos'è Amazon S3 su Outposts?

I seguenti esempi mostrano come utilizzare S3 su Outposts con l'AWS SDK for Java per eseguire e gestire un caricamento in più parti.

Argomenti

- Esecuzione di un caricamento in più parti di un oggetto in un bucket S3 su Outposts
- Copia di un oggetto in un bucket S3 su Outposts tramite un caricamento in più parti
- Elencare le parti di un oggetto in un bucket S3 su Outposts
- Recuperare un elenco di caricamenti in più parti in corso in un bucket S3 su Outposts

Esecuzione di un caricamento in più parti di un oggetto in un bucket S3 su Outposts

L'esempio S3 su Outposts seguente avvia, carica e completa un caricamento in più parti di un oggetto in un bucket utilizzando SDK per Java. Per utilizzare questo esempio, sostituisci *`user input placeholder`* con le tue informazioni. Per ulteriori informazioni, consultare Caricamento di un oggetto utilizzando il caricamento in più parti.

```
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 di un oggetto in un bucket S3 su Outposts tramite un caricamento in più parti

L'esempio seguente S3 su Outposts utilizza SDK per Java per copiare un oggetto in un bucket. Per utilizzare questo esempio, sostituisci *`user input placeholder`* con le tue informazioni. Questo esempio è adattato da Copia di un oggetto utilizzando il caricamento in più parti.

```
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;
    }
}
```

Elencare le parti di un oggetto in un bucket S3 su Outposts

Nell'esempio S3 su Outposts seguente vengono elencate le parti di un oggetto in un bucket utilizzando SDK per Java. Per utilizzare questo esempio, sostituisci *`user input placeholder`* con le tue informazioni.

```
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();
        }
    }
}
```

Recuperare un elenco di caricamenti in più parti in corso in un bucket S3 su Outposts

Nell'esempio S3 su Outposts seguente viene illustrato come recuperare un elenco di caricamenti in più parti in corso da un bucket Outposts utilizzando SDK per Java. Per utilizzare questo esempio, sostituisci *`user input placeholder`* con le tue informazioni. Questo è un esempio adattato dall'esempio della Elenco dei caricamenti in più parti per 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();
        }
```

```
    }
}
```

Utilizzo di URL prefirmati per S3 su Outposts

Per concedere un accesso limitato nel tempo agli oggetti memorizzati in locale su un Outpost senza aggiornare la policy del bucket, puoi utilizzare un URL prefirmato. Con gli URL prefirmati, il proprietario del bucket può condividere oggetti con persone nel suo cloud privato virtuale (VPC) o concedere loro la possibilità di caricare o eliminare oggetti.

Quando crei un URL prefirmato utilizzando gli SDK AWS o l'AWS Command Line Interface (AWS CLI), associ l'URL a un'azione specifica. Puoi concedere un accesso limitato nel tempo all'URL prefirmato anche scegliendo una scadenza personalizzata che può essere di appena 1 secondo e fino a 7 giorni. Quando condividi l'URL prefirmato, l'utente del VPC può eseguire l'azione incorporata nell'URL come se fosse l'utente di firma originale. Una volta raggiunta la scadenza, l'URL non funzionerà più.

Limitazione delle funzionalità degli URL prefirmati

Le funzionalità dell'URL prefirmato sono limitate dalle autorizzazioni dell'utente che lo ha creato. In sostanza, gli URL prefirmati sono token di connessione che consentono l'accesso agli utenti che li possiedono. Pertanto, consigliamo di proteggerli in modo appropriato.

AWS Signature Version 4 (SigV4)

Per applicare un comportamento specifico quando le richieste dell'URL prefirmato vengono autenticate tramite AWS Signature Version 4 (SigV4), puoi utilizzare le chiavi di condizione nelle policy del bucket e nelle policy dei punti di accesso. Ad esempio, puoi creare una policy del bucket che utilizzi la condizione `s3-outposts:signatureAge` per negare qualsiasi richiesta di URL prefirmato da Amazon S3 su Outposts sugli oggetti nel bucket `example-outpost-bucket` se la firma ha più di 10 minuti. Per utilizzare questo comando, sostituisci *`user input placeholders`* con le tue informazioni.

```
{
    "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"}
            }
        }
    ]
}
```

Per un elenco di chiavi di condizione e policy di esempio aggiuntive che è possibile utilizzare per applicare un comportamento specifico quando le richieste dell'URL prefirmato vengono autenticate tramite Signature Version 4, consulta Chiavi di policy specifiche per l'autenticazione con AWS Signature Version 4 (SigV4).

Limitazioni per percorso di rete

Se desideri limitare l'utilizzo di URL prefirmati e tutti gli accessi S3 su Outposts a determinati percorsi di rete, puoi definire policy che richiedono un percorso di rete specifico. Per impostare la restrizione sul principale IAM che effettua la chiamata, puoi utilizzare le policy AWS Identity and Access Management (IAM) basate sull'identità (ad esempio policy di utenti, gruppi o ruoli). Per impostare la restrizione sulla risorsa S3 su Outposts, puoi utilizzare le policy sulle risorse (ad esempio, policy di bucket e punti di accesso).

Una restrizione del percorso di rete sul principale IAM richiede all'utente di tali credenziali di effettuare le richieste dalla rete specificata. Una restrizione sul bucket o sul punto di accesso richiede che tutte le richieste a quella risorsa provengano dalla rete specificata. Queste restrizioni si applicano anche al di fuori dello scenario di URL prefirmato.

La condizione globale IAM utilizzata dipende dal tipo di endpoint. Se utilizzi l'endpoint pubblico per S3 su Outposts, utilizza `aws:SourceIp`. Se utilizzi un endpoint VPC per S3 su Outposts, utilizza `aws:SourceVpc` o `aws:SourceVpce`.

La seguente istruzione di policy IAM richiede che il principale del servizio acceda ad AWS solo dall'intervallo di rete specificato. Con questa istruzione della policy, tutti gli accessi devono avere origine da tale intervallo. Ciò include il caso di un utente che utilizza un URL prefirmato per S3 su Outposts. Per utilizzare questo comando, sostituisci *`user input placeholders`* con le tue informazioni.

```
{
    "Sid": "NetworkRestrictionForIAMPrincipal",
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
        "NotIpAddressIfExists": {"aws:SourceIp": "IP-address-range"},
        "BoolIfExists": {"aws:ViaAWSService": "false"}
    }
}
```

Per un esempio di policy del bucket che utilizza la chiave di condizione globale `aws:SourceIP` AWS per limitare l'accesso a un bucket S3 su Outposts a un intervallo di rete specifico, consulta Configurazione di IAM con S3 su Outposts.

Chi può creare un URL prefirmato

Qualsiasi utente che disponga di credenziali di sicurezza valide può creare un URL prefirmato. Tuttavia, perché un utente nel VPC possa accedere a un oggetto, è necessario che l'URL prefirmato venga creato da un utente che dispone dell'autorizzazione a eseguire l'operazione su cui si basa l'URL prefirmato.

Per creare un URL prefirmato puoi utilizzare le seguenti credenziali:

- Profilo dell'istanza IAM: valido fino a 6 ore.
- AWS Security Token Service: valido fino a 36 ore quando viene firmato con credenziali permanenti, ad esempio quelle dell'utente root dell'Account AWS o di un utente IAM.
- Utente IAM: valido fino a 7 giorni quando utilizzi AWS Signature Version 4.

  Per creare un URL prefirmato valido fino a 7 giorni, devi prima delegare le credenziali dell'utente IAM (la chiave di accesso e la chiave segreta) all'SDK in uso. Quindi, genera un URL prefirmato utilizzando AWS Signature Version 4.

> **Note**
>
> - Se hai creato un URL prefirmato utilizzando un token temporaneo, l'URL scade insieme al token, anche se per l'URL è indicata una data di scadenza successiva.

- Poiché gli URL prefirmati consentono l'accesso ai tuoi bucket S3 su Outposts a chiunque disponga dell'URL, ti consigliamo di proteggere tali URL in modo appropriato. Per ulteriori informazioni sulla protezione di URL prefirmati, consulta la sezione Limitazione delle funzionalità degli URL prefirmati.

Quando S3 su Outposts verifica la data e l'ora di scadenza in un URL prefirmato?

Al momento della richiesta HTTP, S3 su Outposts controlla la data e l'ora di scadenza di un URL firmato. Ad esempio, se un client inizia a scaricare un file di grandi dimensioni immediatamente prima dell'ora di scadenza, il download viene completato anche se l'ora di scadenza viene superata. Se la connessione TCP viene interrotta e il client prova a riavviare il download dopo la scadenza, il download non riesce.

Per ulteriori informazioni sull'utilizzo di un URL prefirmato per condividere o caricare oggetti, consulta gli argomenti riportati di seguito.

Argomenti

- Condivisione di oggetti mediante URL prefirmati
- Generazione di un URL prefirmato per il caricamento di un oggetto in un bucket S3 su Outposts

Condivisione di oggetti mediante URL prefirmati

Per concedere un accesso limitato nel tempo agli oggetti memorizzati in locale su un Outpost senza aggiornare la policy del bucket, puoi utilizzare un URL prefirmato. Con gli URL prefirmati, il proprietario del bucket può condividere oggetti con persone nel suo cloud privato virtuale (VPC) o concedere loro la possibilità di caricare o eliminare oggetti.

Quando crei un URL predefinito utilizzando gli AWS SDK o AWS Command Line Interface (AWS CLI), associ l'URL a un'azione specifica. Puoi concedere un accesso limitato nel tempo all'URL prefirmato anche scegliendo una scadenza personalizzata che può essere di appena 1 secondo e fino a 7 giorni. Quando condividi l'URL prefirmato, l'utente del VPC può eseguire l'azione incorporata nell'URL come se fosse l'utente di firma originale. Una volta raggiunta la scadenza, l'URL non funzionerà più.

Quando crei un URL prefirmato, devi fornire le credenziali di sicurezza e specificare quanto segue:

- Un nome della risorsa Amazon (ARN) del punto di accesso per il bucket S3 su Outposts.
- Una chiave oggetto
- Un metodo HTTP (`GET` per scaricare gli oggetti)
- Una data e un'ora di scadenza

Un URL prefirmato è valido solo per la durata specificata. In altre parole, è necessario avviare l'operazione consentita dall'URL prima della sua data e ora di scadenza. L'URL prefirmato può essere utilizzato più volte, fino alla data e all'ora di scadenza. Se hai creato un URL prefirmato utilizzando un token temporaneo, l'URL scade insieme al token, anche se per l'URL è indicata una data di scadenza successiva.

Gli utenti nel cloud privato virtuale (VPC) che hanno accesso all'URL prefirmato possono caricare oggetti. Ad esempio, se il bucket contiene un video e sia il bucket che l'oggetto sono privati, è possibile condividere il video con altri generando un URL prefirmato. Poiché gli URL prefirmati consentono l'accesso ai tuoi bucket S3 su Outposts a chiunque disponga dell'URL, ti consigliamo di proteggere tali URL in modo appropriato. Per ulteriori informazioni sulla protezione degli URL prefirmati, consulta la sezione Limitazione delle funzionalità degli URL prefirmati.

Qualsiasi utente che disponga di credenziali di sicurezza valide può creare un URL prefirmato. Tuttavia, l'URL prefirmato deve essere creato da un utente dotato dell'autorizzazione per eseguire l'operazione su cui si basa l'URL. Per ulteriori informazioni, consulta Chi può creare un URL prefirmato.

È possibile generare un URL prefirmato per condividere un oggetto in un bucket S3 su Outposts utilizzando AWS SDK e AWS CLI. Per maggiori informazioni, consulta i seguenti esempi.

Utilizzo degli SDK AWS

Puoi utilizzare gli AWS SDK per generare un URL predefinito da fornire ad altri in modo che possano recuperare un oggetto.

> **Note**
>
> Quando utilizzi gli AWS SDK per generare un URL predefinito, il tempo di scadenza massimo per un URL predefinito è di 7 giorni dal momento della creazione.

Java

Example

Nel seguente esempio viene generato un URL prefirmato che è possibile fornire ad altri utenti in modo che possano recuperare un oggetto da un bucket S3 su Outposts. Per ulteriori informazioni, consulta Utilizzo di URL prefirmati per S3 su Outposts. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

Per istruzioni su come creare e testare un esempio funzionante, consulta Getting Started nella Developer Guide. 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

Nel seguente esempio viene generato un URL prefirmato che è possibile fornire ad altri utenti in modo che possano recuperare un oggetto da un bucket S3 su Outposts. Per ulteriori informazioni, consulta Utilizzo di URL prefirmati per S3 su Outposts. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

Per informazioni sulla configurazione e l'esecuzione degli esempi di codice, consulta Getting Started with the AWS SDK for .NET nella AWS SDK for .NET Developer Guide.

```
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

Nel seguente esempio viene generato un URL prefirmato per condividere un oggetto utilizzando l'SDK per Python (Boto3). Ad esempio, utilizza un client Boto3 e la funzione `generate_presigned_url` per generare un URL prefirmato che ti consenta di eseguire il `GET` di un oggetto.

```
import boto3
    url = boto3.client('s3').generate_presigned_url(
    ClientMethod='get_object',
    Params={'Bucket': 'ACCESS_POINT_ARN', 'Key': 'OBJECT_KEY'},
    ExpiresIn=3600)
```

Per ulteriori informazioni sull'utilizzo dell'SDK per Python (Boto3) per generare un URL prefirmato, consulta Python nella Documentazione di riferimento delle API di AWS SDK for Python (Boto) .

Utilizzo del AWS CLI

Il AWS CLI comando di esempio seguente genera un URL predefinito per un bucket S3 on Outposts. Per utilizzare questo comando, sostituisci *`user input placeholders`* con le tue informazioni.

> **Note**
>
> Quando si utilizza il AWS CLI per generare un URL predefinito, il tempo di scadenza massimo per un URL predefinito è di 7 giorni dal momento della creazione.

```
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
```

Per ulteriori informazioni, consulta presign in Riferimento ai comandi della AWS CLI .

Generazione di un URL prefirmato per il caricamento di un oggetto in un bucket S3 su Outposts

Per concedere un accesso limitato nel tempo agli oggetti memorizzati in locale su un Outpost senza aggiornare la policy del bucket, puoi utilizzare un URL prefirmato. Con gli URL prefirmati, il

proprietario del bucket può condividere oggetti con persone nel suo cloud privato virtuale (VPC) o concedere loro la possibilità di caricare o eliminare oggetti.

Quando crei un URL prefirmato utilizzando gli SDK AWS o l'AWS Command Line Interface (AWS CLI), associ l'URL a un'azione specifica. Puoi concedere un accesso limitato nel tempo all'URL prefirmato anche scegliendo una scadenza personalizzata che può essere di appena 1 secondo e fino a 7 giorni. Quando condividi l'URL prefirmato, l'utente del VPC può eseguire l'azione incorporata nell'URL come se fosse l'utente di firma originale. Una volta raggiunta la scadenza, l'URL non funzionerà più.

Quando crei un URL prefirmato, devi fornire le credenziali di sicurezza e specificare quanto segue:

- Un nome della risorsa Amazon (ARN) del punto di accesso per il bucket S3 su Outposts.
- Una chiave oggetto
- Un metodo HTTP (PUT per il caricamento di oggetti)
- Una data e un'ora di scadenza

Un URL prefirmato è valido solo per la durata specificata. In altre parole, è necessario avviare l'operazione consentita dall'URL prima della sua data e ora di scadenza. L'URL prefirmato può essere utilizzato più volte, fino alla data e all'ora di scadenza. Se hai creato un URL prefirmato utilizzando un token temporaneo, l'URL scade insieme al token, anche se per l'URL è indicata una data di scadenza successiva.

Se l'operazione consentita da un URL prefirmato è costituita da più fasi, ad esempio un caricamento in più parti, tutti le fasi devono essere avviate prima della scadenza. Se S3 su Outposts prova ad avviare una fase con un URL scaduto, viene restituito un errore.

Gli utenti nel cloud privato virtuale (VPC) che hanno accesso all'URL prefirmato possono caricare oggetti. Ad esempio, un utente nel VPC che ha accesso all'URL prefirmato può caricare un oggetto nel tuo bucket. Poiché gli URL prefirmati consentono l'accesso ai tuoi bucket S3 su Outposts a qualsiasi utente nel VPC che abbia accesso all'URL prefirmato, ti consigliamo di proteggere tali URL in modo appropriato. Per ulteriori informazioni sulla protezione degli URL prefirmati, consulta la sezione Limitazione delle funzionalità degli URL prefirmati.

Qualsiasi utente che disponga di credenziali di sicurezza valide può creare un URL prefirmato. Tuttavia, l'URL prefirmato deve essere creato da un utente dotato dell'autorizzazione per eseguire l'operazione su cui si basa l'URL. Per ulteriori informazioni, consulta Chi può creare un URL prefirmato.

Uso degli SDK AWS per generare un URL prefirmato per un'operazione sugli oggetti S3 su Outpost

Java

SDK per Java 2.x

Questo esempio mostra come generare un URL prefirmato utilizzabile da un bucket S3 su Outposts per un periodo di tempo limitato. Per ulteriori informazioni, consulta Utilizzo di URL prefirmati per S3 su 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 per Python (Boto3)

In questo esempio viene mostrato come generare un URL prefirmato in grado di eseguire un'operazione S3 su Outposts per un periodo di tempo limitato. Per ulteriori informazioni, consulta Utilizzo di URL prefirmati per S3 su Outposts. Per effettuare una richiesta con l'URL, utilizza il pacchetto `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 su Outposts con Amazon EMR locale su Outposts

Amazon EMR è una piattaforma di cluster gestita che semplifica l'esecuzione di framework di big data, ad esempio eApache Spark, AWS per elaborare Apache Hadoop e analizzare grandi quantità di dati. Utilizzando questi framework e i relativi progetti open source, puoi elaborare i dati per scopi di analisi e carichi di lavoro di business intelligence. Amazon EMR ti aiuta anche a trasformare e spostare grandi quantità di dati da e verso altri archivi di AWS dati e database e supporta Amazon S3 on Outposts. Per ulteriori informazioni su Amazon EMR, consulta Amazon EMR on Outposts nella Amazon EMR Management Guide.

Per Amazon S3 on Outposts, Amazon EMR ha iniziato a supportare il connettore S3A Apache Hadoop nella versione 7.0.0. Le versioni precedenti di Amazon EMR non supportano S3 locale su Outposts e l'EMR File System (EMRFS) non è supportato.

Applicazioni supportate

Amazon EMR con Amazon S3 on Outposts supporta le seguenti applicazioni:

- Hadoop
- Spark
- Hue
- Hive
- Sqoop
- Pig
- Hudi
- Flink

Per ulteriori informazioni, consulta la Guida ai rilasci di Amazon EMR.

Crea e configura un bucket Amazon S3 on Outposts

Amazon EMR utilizza Amazon S3 on Outposts per archiviare dati di input e output. AWS SDK for Java I tuoi file di log di Amazon EMR sono archiviati in una posizione Amazon S3 regionale selezionata e non sono archiviati localmente su Outpost. Per ulteriori informazioni, consulta i log di Amazon EMR nella Amazon EMR Management Guide.

Per soddisfare i requisiti di Amazon S3 e DNS, i bucket S3 on Outposts hanno restrizioni e limitazioni di denominazione. Per ulteriori informazioni, consulta Creazione di un bucket S3 su Outposts.

Con Amazon EMR versione 7.0.0 e successive, puoi utilizzare Amazon EMR con S3 on Outposts e il file system S3A.

Prerequisiti

Autorizzazioni S3 on Outposts: quando crei il tuo profilo di istanza Amazon EMR, il tuo ruolo deve contenere AWS Identity and Access Management lo spazio dei nomi (IAM) per S3 on Outposts. S3 on Outposts ha il proprio spazio dei nomi,. `s3-outposts*` Per un esempio di politica che utilizza questo spazio dei nomi, vedi. Configurazione di IAM con S3 su Outposts

Connettore S3A: per configurare il cluster EMR per accedere ai dati da un bucket Amazon S3 on Outposts, devi utilizzare il connettore S3A. Apache Hadoop Per utilizzare il connettore, assicurati che tutti gli URI S3 utilizzino lo schema. `s3a` In caso contrario, puoi configurare l'implementazione del file system che usi per il tuo cluster EMR in modo che gli URI S3 funzionino con il connettore S3A.

Per configurare l'implementazione del file system in modo che funzioni con il connettore S3A, utilizzi le proprietà `fs.`*`file_scheme`*`.impl` e di `fs.AbstractFileSystem.`*`file_scheme`*`.impl` configurazione per il tuo cluster EMR, dove *`file_scheme`* corrisponde al tipo di URI S3 di cui disponi. Per utilizzare il seguente esempio, sostituisci *`user input placeholders`* con le tue informazioni. Ad esempio, per modificare l'implementazione del file system per gli URI S3 che utilizzano `s3` lo schema, specifica le seguenti proprietà di configurazione del cluster:

```
[
  {
"Classification": "core-site",
    "Properties": {
    "fs.s3.impl": "org.apache.hadoop.fs.s3a.S3AFileSystem",
    "fs.AbstractFileSystem.s3.impl": "org.apache.hadoop.fs.s3a.S3A"
    }
  }
]
```

Per utilizzare S3A, imposta la proprietà di `fs.`*`file_scheme`*`.impl` configurazione su e imposta `org.apache.hadoop.fs.s3a.S3AFileSystem` la proprietà su. `fs.AbstractFileSystem.`*`file_scheme`*`.impl` `org.apache.hadoop.fs.s3a.S3A`

Ad esempio, se state accedendo al percorso`s3a://`*`bucket`*`/...`, impostate la `fs.s3a.impl` proprietà su e impostate la `fs.AbstractFileSystem.s3a.impl` proprietà su. `org.apache.hadoop.fs.s3a.S3AFileSystem` `org.apache.hadoop.fs.s3a.S3A`

Inizia a usare Amazon EMR con Amazon S3 on Outposts

I seguenti argomenti spiegano come iniziare a usare Amazon EMR con Amazon S3 on Outposts.

Argomenti

- Creazione di una policy di autorizzazione
- Creazione e configurazione del cluster
- Panoramica delle configurazioni

- Considerazioni

Creazione di una policy di autorizzazione

Prima di poter creare un cluster EMR che utilizzi Amazon S3 su Outposts, devi creare una policy IAM da collegare al profilo dell'istanza Amazon EC2 per il cluster. La policy deve disporre delle autorizzazioni per accedere al punto di accesso Amazon Resource Name (ARN) di S3 on Outposts. Per ulteriori informazioni sulla creazione di policy IAM per S3 su Outposts, consulta. Configurazione di IAM con S3 su Outposts

La seguente politica di esempio mostra come concedere le autorizzazioni richieste. Dopo avere creato la policy, collegala al ruolo del profilo dell'istanza utilizzato per creare il cluster EMR, come descritto nella sezione the section called “Creazione e configurazione del cluster”. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

```
{
"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:*"
            ]
        }
    ]

 }
```

Creazione e configurazione del cluster

Per creare un cluster che esegua Spark con S3 su Outposts, completa i seguenti passaggi nella console.

Per creare un cluster che funzioni Spark con S3 su Outposts

1. Apri la console di Amazon EMR all'indirizzo https://console.aws.amazon.com/elasticmapreduce/.
2. Nel pannello di navigazione a sinistra, seleziona Cluster.
3. Scegli Create cluster (Crea cluster).

4. Per la versione di Amazon EMR, scegli emr-7.0.0o successiva.
5. Per il pacchetto di applicazioni, scegli Spark interactive. Quindi seleziona tutte le altre applicazioni supportate che desideri includere nel tuo cluster.
6. Per abilitare Amazon S3 su Outposts, inserisci le impostazioni di configurazione.

   Impostazioni di configurazione di esempio

   Per utilizzare le seguenti impostazioni di configurazione di esempio, *`user input placeholders`* sostituiscile con le tue informazioni.

```
[
 {
   "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. Nella sezione Rete, scegli un cloud privato virtuale (VPC) e una sottorete sul rack. AWS Outposts Per ulteriori informazioni su Amazon EMR on Outposts, consulta la pagina dedicata ai cluster EMR nella Amazon EMR Management Guide. AWS Outposts
8. Nella sezione Profilo dell'istanza EC2 per Amazon EMR, scegli il ruolo IAM a cui è allegata la policy di autorizzazione che hai creato in precedenza.
9. Configura le impostazioni rimanenti del cluster, quindi scegli Crea cluster.

Panoramica delle configurazioni

Le tabelle seguenti descrivono S3A e Spark le configurazioni e i valori da specificare per i relativi parametri quando si configura un cluster che utilizza S3 on Outposts con Amazon EMR.

Configurazioni S3A

Parametro	Valore predefinito	Valore richiesto per S3 on Outposts	Spiegazione
`fs.s3a.aws.credentials.provider`	Se non specificato, S3A cercherà S3 nel bucket Region con il nome del bucket Outposts.	L'ARN del punto di accesso del bucket S3 on Outposts	Amazon S3 su Outposts supporta i punti di accesso configurati solo per i virtual private cloud (VPC) come unico mezzo per accedere ai bucket di Outposts.

Parametro	Valore predefinito	Valore richiesto per S3 on Outposts	Spiegazione
`fs.s3a.committer.name`	`file`	`magic`	Magic committer è l'unico committer supportato per S3 su Outposts.
`fs.s3a.select.enabled`	`TRUE`	`FALSE`	S3 Select non è supportato su Outposts.
`JAVA_HOME`	`/usr/lib/jvm/java-8`	`/usr/lib/jvm/java-11-amazon-corretto.x86_64`	S3 on Outposts on Java S3A richiede la versione 11.

Configurazioni Spark

Parametro	Valore predefinito	Valore richiesto per S3 on Outposts	Spiegazione
`spark.sql.sources.fastS3PartitionDiscovery.enabled`	`TRUE`	`FALSE`	S3 on Outposts non supporta la partizione veloce.
`spark.executorEnv.JAVA_HOME`	`/usr/lib/jvm/java-8`	`/usr/lib/jvm/java-11-amazon-corretto.x86_64`	S3 on Outposts su S3A richiede la versione Java 11.

Considerazioni

Quando integri Amazon EMR con i bucket S3 on Outposts, considera quanto segue:

- Amazon S3 on Outposts è supportato con Amazon EMR versione 7.0.0 e successive.
- Il connettore S3A è necessario per utilizzare S3 on Outposts con Amazon EMR. Solo S3A dispone delle funzionalità necessarie per interagire con i bucket S3 on Outposts. Per informazioni sulla configurazione del connettore S3A, consulta Prerequisiti.
- Amazon S3 on Outposts supporta solo la crittografia lato server con chiavi gestite di Amazon S3 (SSE-S3) con Amazon EMR. Per ulteriori informazioni, consulta the section called “Crittografia dei dati”.
- Amazon S3 on Outposts non supporta le scritture con S3A. FileOutputCommitter Le scritture con i bucket S3A FileOutputCommitter su S3 on Outposts generano il seguente errore InvalidStorageClass: La classe di archiviazione specificata non è valida.
- Amazon S3 on Outposts non è supportato con Amazon EMR Serverless o Amazon EMR su EKS.
- I log di Amazon EMR sono archiviati in una posizione Amazon S3 regionale selezionata dall'utente e non sono archiviati localmente nel bucket S3 on Outposts.

Memorizzazione nella cache di autorizzazione e autenticazione

S3 on Outposts memorizza in modo sicuro i dati di autenticazione e autorizzazione localmente sui rack Outposts. La cache rimuove i round trip verso il dispositivo principale per ogni richiesta. Regione AWS Ciò elimina la variabilità introdotta dai round trip di rete. Con la cache di autenticazione e autorizzazione di S3 on Outposts, ottieni latenze coerenti che sono indipendenti dalla latenza della connessione tra Outposts e. Regione AWS

Quando effettui una richiesta API S3 on Outposts, i dati di autenticazione e autorizzazione vengono memorizzati in modo sicuro nella cache. I dati memorizzati nella cache vengono quindi utilizzati per autenticare le successive richieste API degli oggetti S3. S3 on Outposts memorizza nella cache i dati di autenticazione e autorizzazione solo quando la richiesta viene firmata utilizzando Signature Version 4A (SigV4A). La cache è archiviata localmente negli Outposts all'interno del servizio S3 on Outposts. Si aggiorna in modo asincrono quando effettui una richiesta API S3. La cache è crittografata e in Outposts non viene memorizzata alcuna chiave crittografica in testo semplice.

La cache è valida per un massimo di 10 minuti quando Outpost è connesso a. Regione AWS Viene aggiornato in modo asincrono quando effettui una richiesta API S3 on Outposts, per garantire che vengano utilizzate le politiche più recenti. Se Outpost è disconnesso da Regione AWS, la cache sarà valida per un massimo di 12 ore.

Configurazione della cache di autorizzazione e autenticazione

S3 on Outposts memorizza automaticamente nella cache i dati di autenticazione e autorizzazione per le richieste firmate con l'algoritmo SigV4a. Per ulteriori informazioni, consulta Firmare le richieste AWS API nella Guida per l'utente.AWS Identity and Access Management L'algoritmo Sigv4a è disponibile nelle versioni più recenti degli SDK. AWS È possibile ottenerlo tramite una dipendenza dalle librerie AWS Common Runtime (CRT).

È necessario utilizzare la versione più recente dell' AWS SDK e installare l'ultima versione del CRT. Ad esempio, puoi eseguire per `pip install awscrt` ottenere la versione più recente del CRT con Boto3.

S3 on Outposts non memorizza nella cache i dati di autenticazione e autorizzazione per le richieste firmate con l'algoritmo SigV4.

Convalida della firma SigV4A

È possibile utilizzare AWS CloudTrail per convalidare che le richieste siano state firmate con SigV4a. Per ulteriori informazioni sulla configurazione CloudTrail di S3 su Outposts, consulta. Monitoraggio di S3 su Outposts con i log AWS CloudTrail

Dopo la configurazione CloudTrail, puoi verificare come è stata firmata una richiesta nel `SignatureVersion` campo dei CloudTrail log. Le richieste firmate con SigV4a avranno un valore impostato su. `SignatureVersion` `AWS 4-ECDSA-P256-SHA256` Le richieste firmate con SigV4 saranno impostate su. `SignatureVersion` `AWS 4-HMAC-SHA256`

Sicurezza in S3 su Outposts

Per AWS, la sicurezza del cloud ha la massima priorità. In quanto cliente AWS, puoi trarre vantaggio da un'architettura di data center e di rete progettata per soddisfare i requisiti delle aziende più esigenti a livello di sicurezza.

La sicurezza è una responsabilità condivisa tra AWS e l'utente. Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- La sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che esegue Servizi AWS nel Cloud AWS. AWS fornisce, inoltre, servizi utilizzabili in modo sicuro. Revisori di terze parti testano regolarmente e verificano l'efficacia della nostra sicurezza nell'ambito dei Programmi

di conformità AWS. Per informazioni sui programmi di conformità applicabili ad Amazon S3 su Outposts, consulta Servizi AWS coperti dal programma di conformità.

- Sicurezza nel cloud: la tua responsabilità è determinata dal Servizio AWS che viene utilizzato. Inoltre, sei responsabile anche di altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda e le leggi e le normative applicabili.

Questa documentazione illustra come applicare il modello di responsabilità condivisa quando si usa S3 su Outposts. I seguenti argomenti illustrano come configurare S3 su Outposts per soddisfare gli obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri servizi Servizi AWS per monitorare e proteggere le risorse S3 su Outposts.

Argomenti

- Crittografia dei dati in S3 su Outposts
- AWS PrivateLink per S3 su Outposts
- Chiavi di policy specifiche per l'autenticazione con AWS Signature Version 4 (SigV4)
- Policy gestite da AWS per Amazon S3 su Outposts
- Utilizzo dei ruoli collegati ai servizi per Amazon S3 su Outposts

Crittografia dei dati in S3 su Outposts

Per default, tutti i dati memorizzati in Amazon S3 su Outposts vengono crittografati utilizzando la crittografia lato server con chiavi di crittografia gestite di Amazon S3 (SSE-S3). Per ulteriori informazioni, consultare Uso della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).

È possibile specificare la crittografia lato server con chiavi di crittografia fornite dal cliente (SSE-C). Per utilizzare SSE-C, specifica una chiave di crittografia come parte delle richieste API sull'oggetto. La crittografia lato server viene applicata solo ai dati dell'oggetto, non dei metadati dell'oggetto. Per ulteriori informazioni, consultare Utilizzo della crittografia lato server con chiavi fornite dal cliente (SSE-C).

> **Note**
>
> Amazon S3 su Outposts non supporta la crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS).

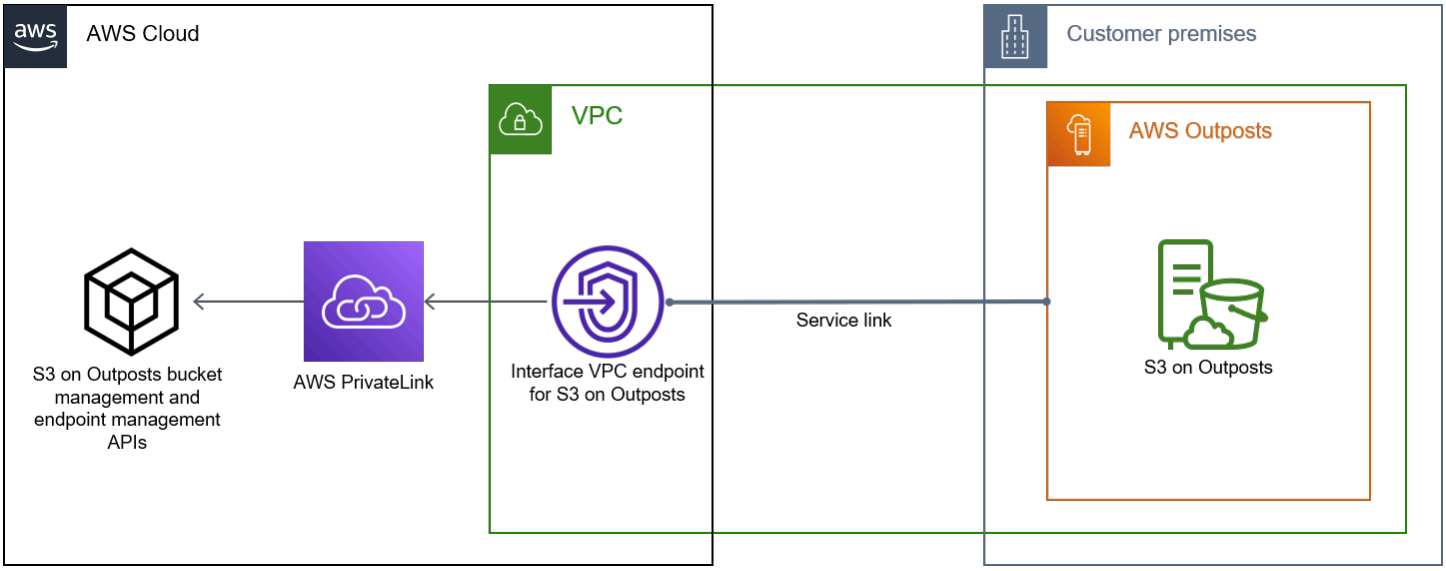
AWS PrivateLink per S3 su Outposts

AWS PrivateLink Supporta S3 on Outposts, che fornisce l'accesso diretto alla gestione dello storage S3 on Outposts tramite un endpoint privato all'interno della rete privata virtuale. Ciò consente di semplificare l'architettura di rete interna ed eseguire operazioni di gestione sullo storage di oggetti Outposts utilizzando indirizzi IP privati nel cloud privato virtuale (VPC). L'utilizzo AWS PrivateLink elimina la necessità di utilizzare indirizzi IP pubblici o server proxy.

Con AWS PrivateLink for Amazon S3 on Outposts, puoi effettuare il provisioning degli endpoint VPC di interfaccia nel tuo cloud privato virtuale (VPC) per accedere alle API di gestione dei bucket e degli endpoint di S3 on Outposts. Gli endpoint VPC dell'interfaccia sono accessibili alle applicazioni distribuite nel VPC o on-premise sulla rete privata virtuale (VPN) o AWS Direct Connect. Puoi accedere alle API di gestione dei bucket e degli endpoint tramite. AWS PrivateLink AWS PrivateLink non supporta operazioni API di trasferimento dati, come GET, PUT e API simili. Queste operazioni vengono già trasferite privatamente tramite la configurazione dell'endpoint e del punto di accesso S3 su Outposts. Per ulteriori informazioni, consulta Reti per S3 su Outposts.

Gli endpoint di interfaccia sono rappresentati da una o più interfacce di rete elastiche (ENI) a cui vengono assegnati indirizzi IP privati dalle sottoreti nel VPC. Le richieste effettuate agli endpoint dell'interfaccia per S3 su Outposts vengono instradate automaticamente alle API di gestione dei bucket e degli endpoint S3 su Outposts sulla rete AWS. Puoi anche accedere agli endpoint di interfaccia nel tuo VPC da applicazioni locali AWS Direct Connect tramite AWS Virtual Private Network o ().AWS VPN Per ulteriori informazioni su come connettere il VPC alla rete On-Premise, consulta la Guida per l'utente di AWS Direct Connect e la Guida per l'utente di AWS Site-to-Site VPN .

Gli endpoint di interfaccia instradano le richieste per S3 sui bucket Outposts e sulle API di gestione degli endpoint attraverso la AWS rete e attraverso AWS PrivateLink, come illustrato nel diagramma seguente.



Per informazioni sulla creazione di endpoint di interfaccia, consulta Endpoint VPC di interfaccia (AWS PrivateLink) nella Guida di AWS PrivateLink .

Argomenti

- Restrizioni e limitazioni
- Accesso a endpoint dell'interfaccia S3 su Outposts
- Aggiornamento di una configurazione DNS locale
- Creazione di un endpoint VPC per S3 su Outposts
- Creazione di policy di bucket e policy di endpoint VPC per S3 su Outposts

Restrizioni e limitazioni

Quando accedi a S3 su bucket Outposts e API di gestione degli endpoint tramite AWS PrivateLink, si applicano le limitazioni del VPC. Per ulteriori informazioni, consulta Proprietà e limitazioni degli endpoint di interfaccia e Quote di AWS PrivateLink nella Guida di AWS PrivateLink .

Inoltre, non supporta quanto segue AWS PrivateLink :

- Endpoint FIPS (Federal Information Processing Standard)
- API di trasferimento dati S3 su Outposts, ad esempio, operazioni GET, PUT e API di oggetti simili.
- DNS privato

Accesso a endpoint dell'interfaccia S3 su Outposts

Per accedere a S3 on Outposts utilizzando le API di gestione degli endpoint e del bucket, devi aggiornare le tue applicazioni per AWS PrivateLink utilizzare nomi DNS specifici degli endpoint. Quando crei un endpoint di interfaccia, AWS PrivateLink genera due tipi di S3 specifici per endpoint sui nomi Outposts: regionale e zonale.

- Nomi DNS regionali: includono un ID endpoint VPC univoco, un identificatore di servizio, `vpce.amazonaws.com` e, Regione AWS ad esempio,. *`vpce-1a2b3c4d-5e6f`*`.s3-outposts.us-east-1.vpce.amazonaws.com`
- Nomi DNS zonali: includono un ID endpoint VPC univoco, la zona di disponibilità, un identificatore di servizio, e, ad esempio Regione AWS,. `vpce.amazonaws.com` *`vpce-1a2b3c4d-5e6f`***`-us-east-1a.`**`s3-outposts.us-east-1.vpce.amazonaws.com` Puoi utilizzare questa opzione se l'architettura isola le zone di disponibilità. Ad esempio, puoi utilizzare i nomi DNS zonali per il contenimento degli errori o per ridurre i costi di trasferimento dei dati a livello regionale.

> **Important**
>
> Gli endpoint dell'interfaccia S3 su Outposts vengono risolti dal dominio DNS pubblico. S3 su Outposts non supporta il DNS privato. Usa il parametro `--endpoint-url` per tutte le API di gestione dei bucket e degli endpoint.

AWS CLI esempi

Utilizzo dei parametri `--region` e `--endpoint-url` per accedere alle API di gestione dei bucket e degli endpoint tramite gli endpoint dell'interfaccia S3 su Outposts.

Example : utilizzo dell'URL dell'endpoint per elencare i bucket con l'API di controllo S3

Nell'esempio seguente, sostituisci la Regione *`us-east-1`*, l'URL endpoint VPC *`vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com`* e l'ID account *`111122223333`* con le informazioni appropriate.

```
aws s3control list-regional-buckets --region us-east-1 --endpoint-url
 https://vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com --account-
id 111122223333
```

AWS esempi SDK

Aggiorna gli SDK alla versione più recente e configura i client per utilizzare un URL endpoint per accedere all'API di controllo S3 per gli endpoint di interfaccia S3 su Outposts. Per ulteriori informazioni, consulta Esempi di SDK AWS per AWS PrivateLink.

SDK for Python (Boto3)

Example : utilizzo di un URL endpoint per accedere all'API di controllo S3

Nell'esempio seguente, sostituisci la Regione *us-east-1* e l'URL endpoint VPC *vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com* con le informazioni appropriate.

```
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'
)
```

Per ulteriori informazioni, consulta AWS PrivateLink per Amazon S3 nella Guida per sviluppatori di Boto3.

SDK for Java 2.x

Example : utilizzo di un URL endpoint per accedere all'API di controllo S3

Nell'esempio seguente, sostituire l'URL endpoint VPC *vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com* e la Regione *Region.US_EAST_1* con le informazioni appropriate.

```
// 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()
```

Per ulteriori informazioni, consulta `S3ControlClient` nella documentazione di riferimento dell'API AWS SDK for Java .

Aggiornamento di una configurazione DNS locale

Quando si utilizzano nomi DNS specifici degli endpoint per accedere agli endpoint di interfaccia per le API di gestione degli endpoint e dei bucket S3 su Outposts, non è necessario aggiornare il resolver DNS locale. Puoi risolvere il nome DNS specifico dell'endpoint con l'indirizzo IP privato dell'endpoint di interfaccia dal dominio DNS di S3 su Outposts pubblico.

Creazione di un endpoint VPC per S3 su Outposts

Per creare un endpoint di interfaccia VPC per S3 su Outposts, vedere Creare un endpoint VPC nella Guida AWS PrivateLink .

Creazione di policy di bucket e policy di endpoint VPC per S3 su Outposts

Puoi allegare una policy di endpoint all'endpoint VPC che controlla l'accesso a S3 su Outposts. Puoi utilizzare la condizione `aws:sourceVpce` nelle policy del bucket S3 su Outposts per limitare l'accesso a bucket specifici da un endpoint VPC specifico. Con le policy degli endpoint VPC, è possibile controllare l'accesso alle API di gestione dei bucket di S3 su Outposts e alle API di gestione degli endpoint. Con le policy dei bucket, è possibile controllare l'accesso alle API di gestione dei bucket S3 su Outposts. Tuttavia, non è possibile gestire l'accesso alle azioni oggetto per S3 su Outposts utilizzando `aws:sourceVpce`.

Le policy di accesso per S3 su Outposts specificano le seguenti informazioni:

- Il principio AWS Identity and Access Management (IAM) per il quale le azioni sono consentite o negate.
- Le operazioni di controllo S3 consentite o rifiutate.
- Le risorse S3 su Outposts su cui le operazioni sono consentite o rifiutate.

Negli esempi seguenti vengono illustrate le policy che limitano l'accesso a un bucket o a un endpoint. Per ulteriori informazioni sulla connettività VPC, consulta le opzioni di connettività da rete a VPC nel white paper Opzioni di connettività AWS Amazon Virtual Private Cloud.

> **Important**
>
> - Quando applichi le policy di esempio per gli endpoint VPC descritte in questa sezione, potresti bloccare involontariamente l'accesso al bucket. Le autorizzazioni del bucket che limitano l'accesso del bucket a connessioni originate dall'endpoint VPC possono bloccare

tutte le connessioni al bucket. Per informazioni su come risolvere questo problema, consulta La policy del bucket ha l'ID del VPC o dell'endpoint VPC sbagliato. Come posso correggere la policy in modo da poter accedere al bucket? nel Knowledge Center di AWS Support .

- Prima di utilizzare le policy di esempio seguenti, sostituire l'ID endpoint VPC con un valore appropriato per il caso d'uso. In caso contrario, non sarà possibile accedere al bucket.
- Se la policy consente l'accesso a un bucket S3 su Outposts da uno specifico endpoint VPC, disabilita l'accesso alla console per quel bucket in quanto le richieste della console non provengono dall'endpoint VPC specificato.

Argomenti

Esempio: limitazione dell'accesso a un bucket specifico da un endpoint VPC

Puoi creare una policy di endpoint che limita l'accesso solo a bucket S3 su Outposts specifici. La seguente politica limita l'accesso per l'azione solo a. GetBucketPolicy *example-outpost-bucket* Per usare questa policy, sostituire i valori di esempio con i propri.

```
{
  "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"
    }
  ]
}
```

Esempio: negazione dell'accesso da un endpoint VPC specifico in una policy del bucket S3 su Outposts

La seguente policy sui bucket di S3 on Outposts GetBucketPolicy nega l'accesso al bucket tramite l'endpoint *example-outpost-bucket* VPC. *vpce-1a2b3c4d*

La condizione `aws:sourceVpce` viene utilizzata per specificare l'endpoint e non richiede un Amazon Resource Name (ARN) per la risorsa dell'endpoint VPC, ma solo l'ID dell'endpoint. Per usare questa policy, sostituire i valori di esempio con i propri.

```
{
    "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"}
            }
        }
    ]
}
```

Chiavi di policy specifiche per l'autenticazione con AWS Signature Version 4 (SigV4)

La tabella seguente mostra le chiavi di condizione relative a AWS Signature Version 4 (SigV4) che puoi utilizzare con Amazon S3 su Outposts. In una policy del bucket, puoi aggiungere queste condizioni per applicare un comportamento specifico quando le richieste vengono autenticate tramite Signature Version 4. Per esempi di policy, consulta Esempi di policy del bucket che utilizzano chiavi di condizione relative a Signature Version 4. Per ulteriori informazioni sull'autenticazione delle richieste tramite Signature Version 4, consulta Autenticazione delle richieste (AWS Signature Version 4) nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Chiavi applicabili per operazioni **`s3-outposts:*`** o una qualsiasi delle operazioni S3 on Outposts

Chiavi applicabili	Descrizione
`s3-outposts:authType`	S3 su Outposts supporta diversi metodi di autenticazione. Per limitare le richieste in arrivo all'utilizzo di un metodo di autenticazione specifico, puoi utilizzare questa chiave di condizione opzionale. Ad esempio, puoi utilizzare questa chiave di condizione per consentire solo l'intestazione `Authorization` HTTP da utilizzare nell'autenticazione della richiesta. Valori validi: `REST-HEADER` `REST-QUERY-STRING`
`s3-outposts:signatureAge`	Il periodo di tempo, espresso in millisecondi, di validità di una firma in una richiesta autenticata. Questa condizione è valida solo per gli URL prefirmati. In Signature Version 4, la chiave di firma è valida per un massimo di sette giorni. Pertanto, anche le firme sono valide per un massimo di sette giorni. Per ulteriori informazioni, consulta Introduzione alla firma delle richieste nella Documentazione di riferimento delle API di Amazon Simple Storage Service. Puoi utilizzare questa condizione per limitare ulteriormente la durata della firma. Valore di esempio: `600000`
`s3-outposts:x-amz-content-sha256`	Questa chiave di condizione può essere utilizzata per non consentire contenuti non firmati nel bucket. Quando utilizzi Signature Version 4, per le richieste che utilizzano l'intestazione `Authorization` , aggiungi l'intestazione `x-amz-content-sha256` nel calcolo della firma e quindi imposti il suo valore sul payload hash.

Chiavi applicabili	Descrizione
	Questa chiave di condizione può essere utilizzata nella policy del bucket per negare qualsiasi caricamento in cui i payload non sono firmati. Ad esempio: • Nega i caricamenti che utilizzano l'intestazione `Authorization` per autenticare le richieste ma non firmare il payload. Per ulteriori informazioni, consulta Trasferimento del carico utile in un unico blocco nella Documentazione di riferimento delle API di Amazon Simple Storage Service. • Nega i caricamenti che utilizzano gli URL prefirmati. Gli URL prefirmati hanno sempre un `UNSIGNED_PAYLOAD` . Per ulteriori informazioni, consulta la sezione Autenticazione delle richieste e Metodi di autenticazione nella Documentazione di riferimento delle API di Amazon Simple Storage Service. Valore valido: `UNSIGNED-PAYLOAD`

Esempi di policy del bucket che utilizzano chiavi di condizione relative a Signature Version 4

Per utilizzare i seguenti esempi, sostituisci *`user input placeholders`* con le tue informazioni.

Example : **`s3-outposts:signatureAge`**

La seguente policy del bucket nega qualsiasi richiesta di URL prefirmato S3 su Outposts sugli oggetti in `example-outpost-bucket` se la firma è più vecchia di 10 minuti.

```
{
    "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 seguente policy del bucket consente solo le richieste che utilizzano l'intestazione `Authorization` per l'autenticazione della richiesta. Qualsiasi richiesta di URL prefirmato verrà negata poiché gli URL prefirmati utilizzano parametri di query per fornire informazioni di richiesta e autenticazione. Per ulteriori informazioni, consulta Metodi di autenticazione nella Documentazione di riferimento delle API di 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 seguente policy del bucket nega qualsiasi caricamento con payload non firmati, ad esempio quelli che utilizzano URL prefirmati. Per ulteriori informazioni, consulta la sezione Autenticazione delle richieste e Metodi di autenticazione nella Documentazione di riferimento delle API di 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"
                     }
               }
         }
   ]
}
```

Policy gestite da AWS per Amazon S3 su Outposts

Una policy gestita da AWS è una policy autonoma creata e amministrata da AWS. Le policy gestite da AWS sono progettate per fornire autorizzazioni per molti casi d'uso comuni in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Ricorda che le policy gestite da AWS potrebbero non concedere autorizzazioni con privilegi minimi per i tuoi casi d'uso specifici perché possono essere utilizzate da tutti i clienti AWS. Consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i tuoi casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle policy gestite da AWS. Se AWS aggiorna le autorizzazioni definite in una policy gestita da AWS, l'aggiornamento riguarda tutte le identità principali (utenti, gruppi e ruoli) a cui è collegata la policy. È molto probabile che AWS aggiorni

una policy gestita da AWS quando viene lanciato un nuovo Servizio AWS o nuove operazioni API diventano disponibili per i servizi esistenti.

Per ulteriori informazioni, consultare Policy gestite da AWS nella Guida per l'utente di IAM.

Policy gestita da AWS: AWSS3OnOutpostsServiceRolePolicy

Aiuta a gestire le risorse di rete per l'utente come parte del ruolo collegato al servizio `AWSServiceRoleForS3OnOutposts`.

Per visualizzare le autorizzazioni per questa policy, consulta AWSS3OnOutpostsServiceRolePolicy.

Aggiornamenti di Amazon S3 su Outposts sulle policy gestite da AWS

Visualizza i dettagli sugli aggiornamenti alle policy gestite da AWS per Amazon S3 su Outposts da quando questo servizio ha iniziato a tenere traccia delle modifiche.

Modifica	Descrizione	Data
S3 su Outposts ha aggiunto `AWSS3OnOutpostsServiceRolePolicy`	S3 su Outposts ha aggiunto `AWSS3OnOutpostsServiceRolePolicy` come parte del ruolo collegato al servizio `AWSServiceRoleForS3OnOutposts` per aiutare a gestire le risorse di rete per conto dell'utente.	3 ottobre 2023
S3 su Outposts ha iniziato a tenere traccia delle modifiche	S3 su Outposts ha iniziato a tenere traccia delle modifiche per le sue policy gestite da AWS.	3 ottobre 2023

Utilizzo dei ruoli collegati ai servizi per Amazon S3 su Outposts

Amazon S3 su Outposts utilizza i ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato al servizio è un tipo di ruolo IAM univoco collegato direttamente a S3

su Outposts. I ruoli collegati ai servizi sono predefiniti da S3 su Outposts e includono tutte le autorizzazioni necessarie al servizio per eseguire chiamate agli altri servizi AWS per tuo conto.

Un ruolo collegato ai servizi semplifica la configurazione di S3 su Outposts perché ti permette di evitare l'aggiunta manuale delle autorizzazioni necessarie. S3 su Outposts definisce le autorizzazioni dei relativi ruoli collegati ai servizi e, salvo diversamente definito, solo S3 su Outposts potrà assumere i propri ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. Questa procedura protegge le risorse di S3 su Outposts perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta Servizi AWS che funzionano con IAM e cerca i servizi che riportano Yes (Sì) nella colonna Service-linked roles (Ruoli collegati ai servizi). Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni del ruolo collegato ai servizi per S3 su Outposts

S3 su Outposts utilizza il ruolo collegato ai servizi denominato AWSServiceRoleForS3OnOutposts per aiutarti a gestire le risorse di rete.

Ai fini dell'assunzione del ruolo, il ruolo collegato ai servizi `AWSServiceRoleForS3OnOutposts` considera attendibili i seguenti servizi:

- `s3-outposts.amazonaws.com`

La policy delle autorizzazioni del ruolo denominata `AWSS3OnOutpostsServiceRolePolicy` consente a S3 su Outposts di eseguire le seguenti operazioni sulle risorse specificate:

```
{
    "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"
        }
    ]
}
```

Per consentire a un'entità IAM (come un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi devi configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato ai servizi per S3 su Outposts

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando crei un endpoint S3 su Outposts nella AWS Management Console, nella AWS CLI o nell'API AWS, S3 su Outposts crea automaticamente il ruolo collegato ai servizi.

Se elimini questo ruolo collegato ai servizi, puoi ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando crei un endpoint S3 su Outposts, esso crea automaticamente il ruolo collegato ai servizi.

Puoi utilizzare la console IAM anche per creare un ruolo collegato ai servizi con il caso d'uso S3 su Outposts. In AWS CLI o in AWS API, crea un ruolo collegato ai servizi con il nome di servizio `s3-outposts.amazonaws.com`. Per ulteriori informazioni, consulta Creazione di un ruolo collegato ai servizi nella Guida per l'utente IAM. Se elimini il ruolo collegato ai servizi, puoi utilizzare lo stesso processo per crearlo nuovamente.

Modifica di un ruolo collegato ai servizi per S3 su Outposts

S3 su Outposts non consente di modificare il ruolo collegato ai servizi `AWSServiceRoleForS3OnOutposts`. Questo include il nome del ruolo perché varie entità possano farvi riferimento. Tuttavia, utilizzando IAM è possibile modificarne la descrizione. Per ulteriori informazioni, consulta Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per S3 su Outposts

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato ai servizi, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato ai servizi prima di poterlo eliminare manualmente.

> **Note**
>
> Se il servizio S3 su Outposts utilizza tale ruolo quando tenti di eliminare le risorse, è possibile che l'eliminazione non riesca. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare le risorse S3 su Outposts utilizzate dal ruolo AWSServiceRoleForS3OnOutposts

1. Elimina gli endpoint S3 su Outposts nel tuo Account AWS in tutte le Regioni AWS.
2. Eliminazione del ruolo collegato ai servizi utilizzando IAM.

   Utilizzare la console IAM, AWS CLI, la AWS o l'API per eliminare i ruoli collegati ai servizi `AWSServiceRoleForS30nOutposts`. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi di S3 su Outposts

S3 su Outposts supporta l'utilizzo di ruoli collegati ai servizi in tutte le Regioni AWS in cui il servizio è disponibile. Per ulteriori informazioni, consulta S3 on Outposts Regions and endpoints.

Gestione dello storage S3 su Outposts

Con Amazon S3 su Outposts è possibile creare bucket S3 su AWS Outposts, nonché archiviare e recuperare facilmente gli oggetti on-Premise per le applicazioni che richiedono l'accesso ai dati in locale, l'elaborazione dei dati in locale e la residenza dei dati. S3 su Outposts fornisce una nuova classe di archiviazione, S3 Outposts (`0UTP0STS`), che utilizza le API Amazon S3 ed è progettata per archiviare i dati in modo durevole e ridondante su più dispositivi e server su AWS Outposts. Comunichi con il bucket Outpost utilizzando un punto di accesso e una connessione di endpoint su un Virtual Private Cloud (VPC). Sui bucket Outposts puoi utilizzare le stesse API e caratteristiche di Amazon S3, comprese policy di accesso, crittografia e tagging. Puoi utilizzare S3 su Outposts

tramite la AWS Management Console, AWS Command Line Interface (AWS CLI), gli SDK AWS o l'API REST. Per ulteriori informazioni, consulta Che cos'è Amazon S3 su Outposts?

Per ulteriori informazioni su come gestire e condividere la capacità di archiviazione di Amazon S3 su Outposts, consulta i seguenti argomenti.

Argomenti

- Gestione del controllo delle versioni S3 per il bucket S3 su Outposts
- Creazione e gestione di una configurazione del ciclo di vita per un bucket Amazon S3 su Outposts
- Replica degli oggetti per S3 su Outposts
- Condivisione di S3 su Outposts utilizzando AWS RAM
- Altri Servizi AWS che utilizzano S3 su Outposts

Gestione del controllo delle versioni S3 per il bucket S3 su Outposts

Se abilitato, il controllo delle versioni S3 conserva più copie distinte di un oggetto nello stesso bucket. Puoi utilizzare il controllo delle versioni S3 per conservare, recuperare e ripristinare qualsiasi versione di ogni oggetto archiviato nei bucket Outposts. Il controllo delle versioni S3 ti consente di eseguire il ripristino a seguito di errori dell'applicazione e operazioni non intenzionali degli utenti.

I bucket Amazon S3 su Outposts hanno tre stati del controllo delle versioni:

- Unversioned (Senza versione): se non hai mai abilitato o sospeso il controllo delle versioni S3 per il tuo bucket, non viene eseguito alcun controllo delle versioni e non viene restituito lo stato del controllo delle versioni S3. Per ulteriori informazioni sulla funzione Controllo versioni S3, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.
- Enabled (Abilitato): il controllo delle versioni S3 è abilitato per gli oggetti nel bucket. A tutti gli oggetti aggiunti al bucket viene assegnato un ID versione univoco. Gli oggetti già esistenti nel bucket al momento dell'attivazione del controllo delle versioni hanno un ID versione `null`. Se modifichi questi o altri oggetti con altre operazioni, come PutObject, i nuovi oggetti ottengono un ID versione univoco.
- Suspended (Sospeso): il controllo delle versioni S3 è sospeso per gli oggetti nel bucket. A tutti gli oggetti aggiunti al bucket dopo la sospensione del controllo delle versioni verrà assegnato l'ID versione `null`. Per ulteriori informazioni, consultare Aggiunta di oggetti a bucket con funzione Controllo delle versioni sospesa.

Dopo aver abilitato il controllo delle versioni S3 per un bucket S3 su Outposts, non è possibile ripristinare lo stato senza versione del bucket. Tuttavia, puoi sospendere il controllo delle versioni. Per ulteriori informazioni sulla funzione Controllo versioni S3, consulta Utilizzo della funzione Controllo delle versioni nei bucket S3.

Per ogni oggetto nel bucket esistono una versione corrente e nessuna o più versioni non correnti. Per ridurre i costi di archiviazione, puoi configurare le regole del ciclo di vita del bucket S3 in modo che le versioni non correnti scadano dopo un periodo di tempo specificato. Per ulteriori informazioni, consultare Creazione e gestione di una configurazione del ciclo di vita per un bucket Amazon S3 su Outposts.

Negli esempi seguenti viene illustrato come abilitare o sospendere il controllo delle versioni per un bucket S3 su Outposts utilizzando la AWS Management Console e AWS Command Line Interface (AWS CLI). Per creare un bucket con il controllo delle versioni S3 abilitato, consulta Creazione di un bucket S3 su Outposts.

> **Note**
>
> L'Account AWS che crea il bucket lo possiede ed è l'unico che può eseguire azioni su di esso. I bucket dispongono di proprietà di configurazione come Outpost, tag, crittografia di default e impostazioni del punto di accesso. Le impostazioni del punto di accesso includono il Virtual Private Cloud (VPC), la policy del punto di accesso per l'accesso agli oggetti nel bucket e altri metadati. Per ulteriori informazioni, consultare Specifiche di S3 su Outposts.

Utilizzo della console S3

Per modificare le impostazioni del controllo delle versioni S3 per il bucket

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Scegli il bucket Outposts per cui desideri abilitare il controllo delle versioni S3.
4. Scegliere la scheda Properties (Proprietà).
5. In Bucket Versioning (Funzione Controllo delle versioni del bucket) scegliere Edit (Modifica).
6. Modifica le impostazioni del controllo delle versioni S3 per il bucket scegliendo una delle seguenti opzioni:

- Per sospendere il controllo delle versioni S3 e interrompere la creazione di nuove versioni per gli oggetti, scegli Suspend (Sospendi).
- Per abilitare il controllo delle versioni S3 e salvare più copie distinte di ciascun oggetto, scegli Enable (Abilita).

7. Scegliere Save changes (Salva modifiche).

Tramite AWS CLI

Per abilitare o sospendere il controllo delle versioni S3 per il bucket utilizzando AWS CLI, usa il comando `put-bucket-versioning` come mostrato negli esempi seguenti. Per usare questi esempi, sostituisci ciascun *`user input placeholder`* con le tue informazioni.

Per ulteriori informazioni, consulta put-bucket-versioning in AWS CLI Reference (Guida di riferimento per i comandi AWS CLI).

Example - Per abilitare il controllo delle versioni 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 - Per sospendere il controllo delle versioni 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
```

Creazione e gestione di una configurazione del ciclo di vita per un bucket Amazon S3 su Outposts

È possibile utilizzare il ciclo di vita S3 per ottimizzare la capacità di archiviazione per Amazon S3 su Outposts. È possibile creare regole del ciclo di vita per far scadere gli oggetti man mano che invecchiano o vengono sostituiti da versioni più recenti. Puoi creare, abilitare, disabilitare o eliminare una regola del ciclo di vita.

Per ulteriori informazioni sul ciclo di vita S3, consulta Gestione del ciclo di vita dello storage.

> **Note**
>
> L'Account AWS che crea il bucket lo possiede ed è l'unico in grado di creare, abilitare, disabilitare o eliminare una regola del ciclo di vita.

Per creare e gestire la configurazione del ciclo di vita per un bucket S3 su Outposts, consulta i seguenti argomenti.

Argomenti

- Creazione e gestione di una regola del ciclo di vita utilizzando AWS Management Console
- Creazione e gestione di una configurazione del ciclo di vita utilizzando AWS CLI e SDK per Java

Creazione e gestione di una regola del ciclo di vita utilizzando AWS Management Console

È possibile utilizzare il ciclo di vita S3 per ottimizzare la capacità di archiviazione per Amazon S3 su Outposts. È possibile creare regole del ciclo di vita per far scadere gli oggetti man mano che invecchiano o vengono sostituiti da versioni più recenti. Puoi creare, abilitare, disabilitare o eliminare una regola del ciclo di vita.

Per ulteriori informazioni sul ciclo di vita S3, consulta Gestione del ciclo di vita dello storage.

> **Note**
>
> L'Account AWS che crea il bucket lo possiede ed è l'unico in grado di creare, abilitare, disabilitare o eliminare una regola del ciclo di vita.

Per creare e gestire una regola del ciclo di vita per S3 su Outposts utilizzando la AWS Management Console, consulta i seguenti argomenti.

Argomenti

- Creazione di una regola del ciclo di vita
- Abilitazione di una regola del ciclo di vita
- Modifica di una regola del ciclo di vita

- Eliminazione di una regola del ciclo di vita

Creazione di una regola del ciclo di vita

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Seleziona il bucket Outposts per il quale desideri creare una regola del ciclo di vita.
4. Seleziona la scheda Gestione, quindi Crea regola ciclo di vita.
5. Inserisci un valore per Lifecycle rule name (Nome della regola del ciclo di vita).
6. In Rule scope (Ambito della regola) scegli una delle opzioni seguenti:

   - Per limitare l'ambito di questa regola a filtri specifici, scegli Limit the scope of this rule using one or more filters (Limita l'ambito di questa regola utilizzando uno o più filtri). Quindi, aggiungi un filtro prefisso, i tag o la dimensione dell'oggetto.
   - Per applicare questa regola del ciclo di vita a tutti gli oggetti del bucket, scegli Apply to all objects in the bucket (Applica a tutti gli oggetti nel bucket).
7. In Lifecycle rule actions (Operazioni regola del ciclo di vita), scegli una delle seguenti opzioni:

   - Expire current versions of objects (Scadenza versioni correnti degli oggetti): per i bucket con il controllo delle versioni abilitato, S3 su Outposts aggiunge un contrassegno di eliminazione e mantiene gli oggetti come versioni non correnti. Per i bucket che non utilizzano il controllo delle versioni S3, S3 su Outposts elimina definitivamente gli oggetti.
   - Permanently delete noncurrent versions of objects (Elimina definitivamente le versioni non correnti degli oggetti): S3 su Outposts elimina definitivamente le versioni non correnti degli oggetti.
   - Delete expired object delete markers or incomplete multipart uploads (Elimina contrassegni di eliminazione oggetti scaduti o caricamenti in più parti incompleti): S3 su Outposts elimina definitivamente i contrassegni di eliminazione degli oggetti scaduti o i caricamenti in più parti incompleti.

     Se limiti l'ambito della regola del ciclo di vita utilizzando i tag degli oggetti, non puoi scegliere Delete expired object delete markers (Elimina i contrassegni di eliminazione degli oggetti scaduti). Inoltre, non puoi scegliere Delete expired object delete markers (Elimina i contrassegni di eliminazione degli oggetti) se scegli Expire current object versions (Scadenza versioni correnti degli oggetti).

> **Note**
>
> I filtri basati sulle dimensioni non possono essere utilizzati con i contrassegni di eliminazione e i caricamenti in più parti incompleti.

8. Se hai scelto Expire current versions of objects (Scadenza versioni correnti degli oggetti) o Permanently delete noncurrent versions of objects (Elimina definitivamente le versioni non correnti degli oggetti), configura il trigger della regola in base a una data specifica o all'età dell'oggetto.
9. Se hai scelto Delete expired object delete markers (Elimina i contrassegni di eliminazione degli oggetti scaduti), per confermare che desideri eseguire l'operazione di eliminazione, seleziona Delete expired object delete markers (Elimina i contrassegni di eliminazione degli oggetti scaduti).
10. In Timeline Summary (Riepilogo della cronologia), rivedi la regola del ciclo di vita e scegli Create rule (Crea regola).

Abilitazione di una regola del ciclo di vita

Per abilitare o disabilitare una regola del ciclo di vita del bucket

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Seleziona il bucket Outposts per cui desideri abilitare o disabilitare una regola del ciclo di vita.
4. Seleziona la scheda Management (Gestione), quindi in Lifecycle rule (Regola del ciclo di vita) scegli la regola del ciclo di vita che desideri abilitare o disabilitare.
5. Per Azione, scegli Abilita o disabilita regola.

Modifica di una regola del ciclo di vita

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Seleziona il bucket Outposts per il quale desideri modificare una regola del ciclo di vita.
4. Seleziona la scheda Gestione e scegli la regola del ciclo di vita che desideri modificare.
5. (Facoltativo) Aggiorna il valore in Lifecycle rule name (Nome della regola del ciclo di vita).

6. In Rule scope (Ambito della regola), modifica l'ambito secondo le necessità:

   - Per limitare l'ambito di questa regola a filtri specifici, scegli Limit the scope of this rule using one or more filters (Limita l'ambito di questa regola utilizzando uno o più filtri). Quindi, aggiungi un filtro prefisso, i tag o la dimensione dell'oggetto.
   - Per applicare questa regola del ciclo di vita a tutti gli oggetti del bucket, scegli Apply to all objects in the bucket (Applica a tutti gli oggetti nel bucket).

7. In Lifecycle rule actions (Operazioni regola del ciclo di vita), scegli una delle seguenti opzioni:

   - Expire current versions of objects (Scadenza versioni correnti degli oggetti): per i bucket con il controllo delle versioni abilitato, S3 su Outposts aggiunge un contrassegno di eliminazione e mantiene gli oggetti come versioni non correnti. Per i bucket che non utilizzano il controllo delle versioni S3, S3 su Outposts elimina definitivamente gli oggetti.
   - Permanently delete noncurrent versions of objects (Elimina definitivamente le versioni non correnti degli oggetti): S3 su Outposts elimina definitivamente le versioni non correnti degli oggetti.
   - Delete expired object delete markers or incomplete multipart uploads (Elimina contrassegni di eliminazione oggetti scaduti o caricamenti in più parti incompleti): S3 su Outposts elimina definitivamente i contrassegni di eliminazione degli oggetti scaduti o i caricamenti in più parti incompleti.

     Se limiti l'ambito della regola del ciclo di vita utilizzando i tag degli oggetti, non puoi scegliere Delete expired object delete markers (Elimina i contrassegni di eliminazione degli oggetti scaduti). Inoltre, non puoi scegliere Delete expired object delete markers (Elimina i contrassegni di eliminazione degli oggetti) se scegli Expire current object versions (Scadenza versioni correnti degli oggetti).

   > **Note**
   >
   > I filtri basati sulle dimensioni non possono essere utilizzati con i contrassegni di eliminazione e i caricamenti in più parti incompleti.

8. Se hai scelto Expire current versions of objects (Scadenza versioni correnti degli oggetti) o Permanently delete noncurrent versions of objects (Elimina definitivamente le versioni non correnti degli oggetti), configura il trigger della regola in base a una data specifica o all'età dell'oggetto.

9. Se hai scelto Delete expired object delete markers (Elimina i contrassegni di eliminazione degli oggetti scaduti), per confermare che desideri eseguire l'operazione di eliminazione, seleziona Delete expired object delete markers (Elimina i contrassegni di eliminazione degli oggetti scaduti).
10. Seleziona Salva.

Eliminazione di una regola del ciclo di vita

1. Apri la console Amazon S3 su https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Seleziona il bucket Outposts per il quale desideri eliminare una regola del ciclo di vita.
4. Seleziona la scheda Management (Gestione), quindi in Lifecycle rule (Regola del ciclo di vita) scegli la regola del ciclo di vita che desideri eliminare.
5. Scegliere Delete (Elimina).

Creazione e gestione di una configurazione del ciclo di vita utilizzando AWS CLI e SDK per Java

È possibile utilizzare il ciclo di vita S3 per ottimizzare la capacità di archiviazione per Amazon S3 su Outposts. È possibile creare regole del ciclo di vita per far scadere gli oggetti man mano che invecchiano o vengono sostituiti da versioni più recenti. Puoi creare, abilitare, disabilitare o eliminare una regola del ciclo di vita.

Per ulteriori informazioni sul ciclo di vita S3, consulta Gestione del ciclo di vita dello storage.

> **Note**
>
> L'Account AWS che crea il bucket lo possiede ed è l'unico in grado di creare, abilitare, disabilitare o eliminare una regola del ciclo di vita.

Per creare e gestire una configurazione del ciclo di vita per un bucket S3 su Outposts utilizzando AWS Command Line Interface (AWS CLI) e AWS SDK for Java, consulta i seguenti esempi.

Argomenti

- PUT di una configurazione del ciclo di vita

- GET di una configurazione del ciclo di vita in un bucket S3 su Outposts

PUT di una configurazione del ciclo di vita

AWS CLI

Nel seguente esempio della AWS CLI viene inserito una policy di configurazione del ciclo di vita in un bucket Outposts. Questa policy specifica che tutti gli oggetti con il prefisso contrassegnato (*myprefix*) e i tag scadono dopo 10 giorni. Per utilizzare questo comando, sostituisci *user input placeholder* con le tue informazioni.

1. Salva la policy di configurazione del ciclo di vita in un file JSON. In questo esempio, il file è denominato `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. Inviare il file JSON come parte del comando CLI `put-bucket-lifecycle-configuration`. Per usare questo comando, sostituire *`user input placeholder`* con le proprie informazioni. Per ulteriori informazioni su questo comando, consulta put-bucket-lifecycle-configuration nella Guida di riferimento di 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

Nel seguente esempio di SDK for Java viene inserita una policy di configurazione del ciclo di vita in un bucket Outposts. La configurazione del ciclo di vita specifica che tutti gli oggetti con il prefisso contrassegnato (*`myprefix`*) e i tag scadono dopo 10 giorni. Per utilizzare questo comando, sostituisci *`user input placeholder`* con le tue informazioni. Per ulteriori informazioni, consulta l'argomento relativo a PutBucketLifecycleConfiguration nella Documentazione di riferimento dell'API di 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());

}
```

GET di una configurazione del ciclo di vita in un bucket S3 su Outposts

AWS CLI

Nel seguente esempio della AWS CLI viene ottenuta una configurazione del ciclo di vita in un bucket Outposts. Per usare questo comando, sostituire *user input placeholder* con le proprie informazioni. Per ulteriori informazioni su questo comando, consulta get-bucket-lifecycle-configuration nella Guida di riferimento di 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

Nel seguente esempio di SDK for Java viene ottenuta una configurazione del ciclo di vita per un bucket Outposts. Per ulteriori informazioni, consulta GetBucketLifecycleConfiguration nella Documentazione di riferimento delle API di 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());

}
```

Replica degli oggetti per S3 su Outposts

Se la funzionalità Replica Amazon S3 è impostata su AWS Outposts, puoi configurare Amazon S3 su Outposts in modo che esegua automaticamente la replica degli oggetti S3 su outpost diversi o tra bucket nello stesso outpost. Puoi utilizzare Replica Amazon S3 su Outposts per gestire più repliche dei tuoi dati nello stesso outpost o in outpost diversi oppure in account diversi, in modo conforme ai requisiti di residenza dei dati. Replica Amazon S3 su Outposts consente di potenziare i tuoi requisiti di conformità dell'archiviazione e la condivisione dei dati tra account. Se devi essere sicuro che le replica siano identiche ai dati di origine, puoi utilizzare Replica Amazon S3 su Outposts per creare repliche degli oggetti che mantengono tutti i metadati, ad esempio l'ora di creazione dell'oggetto originale, i tag e gli ID versione.

Replica Amazon S3 su Outposts fornisce anche metriche e notifiche dettagliate per monitorare lo stato della replica degli oggetti tra bucket. Puoi utilizzare Amazon CloudWatch per monitorare l'avanzamento della replica monitorando i byte in attesa di replica, le operazioni in attesa di replica e la latenza di replica tra i bucket di origine e destinazione. Per diagnosticare e correggere rapidamente i problemi di configurazione, puoi anche configurare Amazon EventBridge in modo da ricevere notifiche relative agli errori di replica degli oggetti. Per ulteriori informazioni, consulta Gestione della replica.

Argomenti

- Configurazione di replica
- Requisiti di Replica Amazon S3 su Outposts
- Elementi replicati
- Elementi non replicati

- Cosa non è supportato da Replica Amazon S3 su Outposts?
- Impostazione della replica
- Gestione della replica

Configurazione di replica

S3 su Outposts archivia la configurazione della replica come file XML. Nel file XML della configurazione della replica è necessario specificare un ruolo AWS Identity and Access Management (IAM) e una o più regole.

```
<ReplicationConfiguration>
    <Role>IAM-role-ARN</Role>
    <Rule>
        ...
    </Rule>
    <Rule>
         ...
    </Rule>
     ...
</ReplicationConfiguration>
```

S3 su Outposts non può replicare oggetti senza la tua autorizzazione. Le autorizzazioni S3 su Outposts vengono concesse con il ruolo IAM specificato nella configurazione della replica. S3 su Outposts assume il ruolo IAM per replicare gli oggetti per tuo conto. È necessario concedere le autorizzazioni necessarie per il ruolo IAM prima di avviare la replica. Per ulteriori informazioni su queste autorizzazioni per S3 su Outposts, consulta Creazione di un ruolo IAM.

Puoi aggiungere una regola a una configurazione di replica quando:

- Vuoi replicare tutti gli oggetti.
- Vuoi replicare un sottoinsieme di oggetti. Identifichi il sottoinsieme di oggetti aggiungendo un filtro alla regola. Nel filtro specifichi un prefisso di chiave o tag dell'oggetto o una combinazione di questi elementi, per identificare il sottoinsieme di oggetti a cui si applica la regola.

Per replicare più sottoinsiemi di oggetti, aggiungi diverse regole a una configurazione di replica. In ogni regola puoi specificare un filtro tramite cui selezionare un particolare sottoinsieme di oggetti. Puoi ad esempio scegliere di replicare gli oggetti con prefissi della chiave `tax/` o `document/`. Per

fare ciò devi aggiungere due regole: una che specifica il filtro prefisso della chiave `tax/` e un'altra che specifica il prefisso della chiave `document/`.

Per ulteriori informazioni sulla configurazione e sulle regole per la replica in S3 su Outposts, consulta ReplicationConfiguration nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Requisiti di Replica Amazon S3 su Outposts

La replica richiede quanto segue:

- L'intervallo CIDR Outpost di destinazione deve essere associato alla tabella della sottorete Outpost di origine. Per ulteriori informazioni, consulta Prerequisiti per la creazione delle regole di replica.
- Sia per il bucket di origine che per quello di destinazione deve essere abilitata la funzione S3 di controllo delle versioni. Per ulteriori informazioni sulla funzione Controllo delle versioni, consulta Gestione del controllo delle versioni S3 per il bucket S3 su Outposts.
- Amazon S3 su Outposts deve disporre dell'autorizzazione necessaria per replicare gli oggetti dal bucket di origine a quello di destinazione per tuo conto. Ciò significa che devi creare un ruolo di servizio da delegare `GET` e `PUT` le autorizzazioni a S3 su Outposts.
  1. Prima di creare il ruolo di servizio, è necessario disporre dell'autorizzazione `GET` sul bucket di origine e dell'autorizzazione `PUT` sul bucket di destinazione.
  2. Per creare il ruolo di servizio per delegare le autorizzazioni a S3 su Outposts, devi prima configurare le autorizzazioni per consentire a un'entità IAM (un utente o un ruolo) di eseguire le operazioni `iam:CreateRole` e `iam:PassRole`. Autorizza quindi l'entità IAM a creare il ruolo di servizio. Per fare in modo che S3 su Outposts assuma il ruolo di servizio per tuo conto e deleghi le autorizzazioni `GET` e `PUT` a S3 su Outposts, devi assegnare le necessarie policy di affidabilità e autorizzazione al ruolo. Per ulteriori informazioni su queste autorizzazioni per S3 su Outposts, consulta Creazione di un ruolo IAM. Per ulteriori informazioni sulla creazione di un ruolo di servizio, consulta la sezione relativa alla creazione di un ruolo di servizio.

Elementi replicati

Per impostazione predefinita, S3 su Outposts replica quanto segue:

- Oggetti creati dopo l'aggiunta di una configurazione di replica.

- Metadati dell'oggetto dagli oggetti di origine alle repliche. Per informazioni su come replicare i metadati dalle repliche agli oggetti di origine, consulta Stato della replica se su Outposts è abilitata la sincronizzazione della modifica della replica Amazon S3.
- Eventuali tag degli oggetti.

Effetto delle operazioni di eliminazione sulla replica

Se si elimina un oggetto dal bucket di origine, per impostazione predefinita si verificano le seguenti azioni:

- Se effettui una richiesta `DELETE` senza specificare l'ID versione dell'oggetto, S3 su Outposts aggiunge un contrassegno di eliminazione. S3 su Outposts gestisce il contrassegno di eliminazione in questo modo:
  - S3 su Outposts non replica il contrassegno di eliminazione per impostazione predefinita.
  - Tuttavia, puoi aggiungere la replica del contrassegno di eliminazione a regole non basate su tag. Per ulteriori informazioni su come abilitare la replica dei contrassegni di eliminazione nella configurazione della replica, consulta Utilizzo della console S3.
- Se in una richiesta `DELETE` specifichi l'ID versione di un oggetto da eliminare, S3 su Outposts elimina la versione dell'oggetto nel bucket di origine. Non viene tuttavia eseguita la replica dell'eliminazione nei bucket di destinazione. In altre parole, non elimina la stessa versione dell'oggetto dai bucket di destinazione. Questo comportamento permette di proteggere i dati da eliminazioni da parte di utenti malintenzionati.

Elementi non replicati

Per impostazione predefinita, S3 su Outposts non replica quanto segue:

- Gli oggetti nel bucket di origine che sono repliche create da un'altra regola di replica. Supponiamo, per esempio, di configurare una replica dove il bucket A è l'origine e il bucket B è la destinazione. Supponiamo ora di aggiungere un'altra configurazione di replica dove il bucket B è l'origine e il bucket C è la destinazione. In questo caso, gli oggetti nel bucket B che sono repliche di oggetti nel bucket A non vengono replicati nel bucket C.
- Oggetti nel bucket di origine che sono già stati replicati in una destinazione diversa. Se, ad esempio, modifichi il bucket di destinazione in una configurazione della replica esistente, S3 su Outposts non replica di nuovo gli oggetti.
- Oggetti creati con crittografia lato server con chiavi di crittografia fornite dai clienti (SSE-C).

- Aggiornamenti alle risorse secondarie a livello di bucket.

  Se, ad esempio, modifichi la configurazione del ciclo di vita o aggiungi una configurazione di notifica nel bucket di origine, tali modifiche non vengono applicate nel bucket di destinazione. Questa funzionalità permette la presenza di configurazioni diverse nei bucket di origine e di destinazione.

- Operazioni eseguite dalla configurazione del ciclo di vita.

  Ad esempio, se abiliti una configurazione del ciclo di vita sul bucket di origine e configuri le operazioni di scadenza, S3 su Outposts crea i contrassegni di eliminazione per gli oggetti scaduti nel bucket di origine, ma non replica gli stessi contrassegni nei bucket di destinazione. Per applicare al bucket di origine e a quello di destinazione la stessa configurazione del ciclo di vita, è sufficiente abilitare quest'ultima in entrambi. Per ulteriori informazioni sulla configurazione del ciclo di vita, consulta Gestione del ciclo di vita dello storage.

Cosa non è supportato da Replica Amazon S3 su Outposts?

Le seguenti funzionalità Amazon S3 non sono al momento supportate da S3 su Outposts.

- S3 Replication Time Control (S3 RTC). Il controllo del tempo di replica di S3 (S3 RTC) non è supportato perché il traffico di oggetti in Replica Amazon S3 su Outposts viene gestito dalla rete locale (gateway locale). Per informazioni sui gateway locali, consulta la pagina relativa all'utilizzo dei gateway locali nella Guida per l'utente di AWS Outposts.
- Replica S3 per le operazioni in batch.

Impostazione della replica

> **Note**
>
> Gli oggetti esistenti nel bucket prima della configurazione della regola di replica non vengono replicati automaticamente. In altre parole, Amazon S3 su Outposts non esegue la replica retroattiva di oggetti. Per replicare gli oggetti creati prima della configurazione della replica, puoi utilizzare l'operazione API `CopyObject` per copiarli nello stesso bucket. Dopo la copia, gli oggetti vengono visualizzati come "nuovi" nel bucket e quindi viene loro applicata la configurazione della replica. Per ulteriori informazioni sulla copia di un oggetto, consulta Copia di un oggetto in un bucket Amazon S3 su Outposts utilizzando AWS SDK for Java

e CopyObject nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Per abilitare la Replica S3 su Outposts, aggiungi una regola di replica al bucket Outposts di origine. La regola di replica indica a S3 su Outposts di replicare gli oggetti come specificato. Nella regola di replica devi fornire le informazioni seguenti:

- Il punto di accesso del bucket Outposts di origine: il nome della risorsa Amazon (ARN) del punto di accesso o l'alias del punto di accesso del bucket dal quale desideri che S3 su Outposts replichi gli oggetti. Per ulteriori informazioni, consulta Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3 su Outposts.
- Gli oggetti da replicare: puoi replicare tutti gli oggetti presenti nel bucket Outposts di origine o solo una parte di essi. Puoi identificare un sottoinsieme specificando nella configurazione un prefisso di nome di chiave, uno o più tag di oggetti oppure entrambi.

  Se, ad esempio, configuri una regola di replica per replicare solo gli oggetti con il prefisso di nome di chiave `Tax/`, S3 su Outposts replica gli oggetti con chiavi come `Tax/doc1` o `Tax/doc2`. Ma non replica un oggetto con la chiave `Legal/doc3`. Se specifichi sia un prefisso sia uno o più tag, S3 su Outposts replica solo gli oggetti con il prefisso della chiave e i tag specificati.
- Il bucket Outposts di destinazione: l'ARN o l'alias del punto di accesso del bucket in cui desideri che S3 su Outposts replichi gli oggetti.

Puoi configurare la regola di replica utilizzando la REST API, gli SDK AWS, AWS Command Line Interface (AWS CLI) o la console di Amazon S3.

S3 su Outposts fornisce anche le operazioni API per supportare la configurazione delle regole di replica. Per ulteriori informazioni, consulta i seguenti argomenti nella Documentazione di riferimento delle API di Amazon Simple Storage Service:

- PutBucketReplication
- GetBucketReplication
- DeleteBucketReplication

Argomenti

- Prerequisiti per la creazione delle regole di replica

- Creazione delle regole di replica su Outposts

Prerequisiti per la creazione delle regole di replica

Argomenti

- Connessione delle sottoreti Outpost di origine e destinazione
- Creazione di un ruolo IAM

Connessione delle sottoreti Outpost di origine e destinazione

Affinché il traffico di replica passi dall'Outpost di origine all'Outpost di destinazione tramite il gateway locale, è necessario aggiungere un nuovo percorso per configurare la rete. È necessario connettere gli intervalli di rete dell'instradamento interdominio senza classi (CIDR) dei punti di accesso. Per ogni coppia di punti di accesso, devi configurare questa connessione una sola volta.

Alcuni passaggi per configurare la connessione variano a seconda del tipo di accesso degli endpoint Outposts associati ai punti di accesso. Il tipo di accesso per gli endpoint è Privato (instradamento diretto del cloud privato virtuale [VPC] per AWS Outposts) o IP di proprietà del cliente (un pool di indirizzi IP di proprietà del cliente [pool CoIP] all'interno della rete on-premise).

Passaggio 1: Trovare l'intervallo CIDR dell'endpoint Outposts di origine

Per trovare l'intervallo CIDR dell'endpoint di origine associato al punto di accesso di origine

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Nell'elenco Bucket di Outposts scegli il bucket di origine che desideri per la replica.
4. Scegli la scheda Punti di accesso di Outposts e scegli il punto di accesso di Outposts per il bucket di origine della tua regola di replica.
5. Scegli l'endpoint di Outposts.
6. Copia l'ID della sottorete da utilizzare nel passaggio 5.
7. Il metodo utilizzato per trovare l'intervallo CIDR dell'endpoint di Outposts di origine dipende dal tipo di accesso dell'endpoint.

   Nella sezione Panoramica dell'endpoint Outposts, esamina il tipo di accesso.

- Se il tipo di accesso è Privato, copia il valore Instradamento interdominio senza classi (CIDR) da utilizzare nel passaggio 6.
- Se il tipo di accesso è IP di proprietà del cliente, procedi come segue:
  1. Copia il valore del pool IPv4 di proprietà del cliente da utilizzare in seguito come ID del pool di indirizzi.
  2. Apri la console AWS Outposts all'indirizzo https://console.aws.amazon.com/outposts/.
  3. Nel riquadro di navigazione, seleziona Tabelle di routing del gateway locale.
  4. Scegli il valore ID tabella di routing del gateway locale dell'Outpost di origine.
  5. Nel riquadro dei dettagli, scegli la scheda Pool CoIP. Incolla il valore dell'ID del pool CoIP che hai copiato in precedenza nella casella di ricerca.
  6. Per il pool CoIP corrispondente, copia il valore CIDR corrispondente dell'endpoint Outposts di origine per utilizzarlo nel passaggio 6.

Passaggio 2: trovare l'ID della sottorete e l'intervallo CIDR dell'endpoint Outposts di destinazione

Per trovare l'ID della sottorete e l'intervallo CIDR dell'endpoint di destinazione associato al punto di accesso di destinazione, segui gli stessi passaggi del passaggio 1 e modifica l'endpoint Outposts di origine con l'endpoint Outposts di destinazione quando esegui i passaggi. Copia il valore dell'ID della sottorete dell'endpoint Outposts di destinazione per utilizzarlo nel passaggio 6. Copia il valore CIDR dell'endpoint Outposts di destinazione per utilizzarlo nel passaggio 5.

Passaggio 3: trovare l'ID del gateway locale dell'Outpost di origine

Per trovare l'ID del gateway locale dell'Outpost di origine

1. Apri la console AWS Outposts all'indirizzo https://console.aws.amazon.com/outposts/.
2. Nel riquadro di navigazione a sinistra scegli Gateway locali.
3. Nella pagina Gateway locali trova l'ID dell'Outpost di origine che desideri utilizzare per la replica.
4. Copia il valore dell'ID del gateway locale dell'Outpost di origine per utilizzarlo nel passaggio 5.

Per informazioni sui gateway locali, consulta Gateway locale nella Guida per l'utente di AWS Outposts.

Passaggio 4: trovare l'ID del gateway locale dell'Outpost di destinazione

Per trovare l'ID del gateway locale dell'Outpost di destinazione, segui gli stessi passaggi del passaggio 3, esclusa la ricerca dell'ID dell'Outpost di destinazione. Copia il valore dell'ID del gateway locale dell'Outpost di destinazione per utilizzarlo nel passaggio 6.

Passaggio 5: configurare la connessione dalla sottorete Outpost di origine alla sottorete Outpost di destinazione

Per configurare la connessione dalla sottorete Outpost di origine alla sottorete Outpost di destinazione

1. Accedi alla AWS Management Console e apri la console VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Nel pannello di navigazione a sinistra, seleziona Sottoreti.
3. Nella casella di ricerca, inserisci l'ID della sottorete per l'endpoint Outposts di origine che hai individuato nel passaggio 1. Scegli la sottorete con l'ID corrispondente.
4. Per l'elemento della sottorete corrispondente, scegli il valore della Tabella di instradamento di questa sottorete.
5. Nella pagina con una tabella di instradamento selezionata, scegli Operazioni e quindi Modifica instradamenti.
6. Nella scheda Modifica instradamenti scegli Aggiungi routing.
7. In Destinazione, inserisci l'intervallo CIDR dell'endpoint Outposts di destinazione che hai individuato nel passaggio 2.
8. In Destinazione, scegli Gateway locale outpost e inserisci l'ID del gateway locale dell'Outpost di origine che hai individuato nel passaggio 3.
9. Seleziona Salva modifiche.
10. Assicurati che lo Stato dell'instradamento sia Attivo.

Passaggio 6: configurare la connessione dalla sottorete Outpost di destinazione alla sottorete Outpost di origine

1. Accedi alla AWS Management Console e apri la console VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Nel pannello di navigazione a sinistra, seleziona Sottoreti.

3. Nella casella di ricerca, inserisci l'ID della sottorete per l'endpoint Outposts di destinazione che hai individuato nel passaggio 2. Scegli la sottorete con l'ID corrispondente.
4. Per l'elemento della sottorete corrispondente, scegli il valore della Tabella di instradamento di questa sottorete.
5. Nella pagina con una tabella di instradamento selezionata, scegli Operazioni e quindi Modifica instradamenti.
6. Nella scheda Modifica instradamenti scegli Aggiungi routing.
7. In Destinazione, inserisci l'intervallo CIDR dell'endpoint Outposts di origine che hai individuato nel passaggio 1.
8. In Destinazione, scegli Gateway locale outpost e inserisci l'ID del gateway locale dell'Outpost di destinazione che hai individuato nel passaggio 4.
9. Seleziona Salva modifiche.
10. Assicurati che lo Stato dell'instradamento sia Attivo.

Dopo aver collegato gli intervalli di rete CIDR dei punti di accesso di origine e di destinazione, è necessario creare un ruolo AWS Identity and Access Management (IAM).

Creazione di un ruolo IAM

Per impostazione predefinita, tutte le risorse S3 su Outposts, ossia bucket, oggetti e risorse secondarie correlate, sono private e solo il proprietario vi può accedere. S3 su Outposts ha bisogno delle autorizzazioni per leggere e replicare gli oggetti dal bucket Outposts di origine. Queste autorizzazioni vengono concesse creando un ruolo del servizio IAM e specificandolo nella configurazione della replica.

In questa sezione vengono illustrate la policy di trust e la policy di autorizzazione minima richiesta. Le procedure dettagliate di esempio forniscono istruzioni dettagliate per la creazione di un ruolo IAM. Per ulteriori informazioni, consulta Creazione delle regole di replica su Outposts. Per ulteriori informazioni sui ruoli IAM, consulta Ruoli IAM nella Guida per l'utente di IAM.

- Di seguito viene mostrata una policy di attendibilità in cui identifichi S3 su Outposts come principale del servizio che può assumere il ruolo.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
```

```
         "Effect":"Allow",
         "Principal":{
            "Service":"s3-outposts.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

- Di seguito viene mostrata una policy di accesso in cui concedi al ruolo le autorizzazioni per eseguire attività di replica per tuo conto. Quando S3 su Outposts assume il ruolo, dispone delle autorizzazioni che sono state specificate in questa policy. Per utilizzare questa policy, sostituisci *`user input placeholders`* con le tue specifiche informazioni. Assicurati di sostituirle con gli ID degli Outpost di origine e di destinazione, i nomi dei bucket e i nomi dei punti di accesso dei bucket Outposts di origine e di destinazione.

```
{
   "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 policy di accesso concede le autorizzazioni per le seguenti operazioni:

- `s3-outposts:GetObjectVersionForReplication`: l'autorizzazione per questa operazione viene concessa a tutti gli oggetti per consentire a S3 su Outposts di ottenere una versione specifica dell'oggetto associata a ciascun oggetto.

- `s3-outposts:GetObjectVersionTagging`: l'autorizzazione per questa operazione sugli oggetti nel bucket *`SOURCE-OUTPOSTS-BUCKET`* (bucket di origine) permettono a S3 su Outposts di leggere i tag degli oggetti per la replica. Per ulteriori informazioni, consulta Aggiunta di tag per bucket S3 su Outposts. Se S3 su Outposts non dispone di questa autorizzazione, replica gli oggetti ma non i relativi tag.
- `s3-outposts:ReplicateObject` e `s3-outposts:ReplicateDelete`: le autorizzazioni per queste operazioni sugli oggetti nel bucket *`DESTINATION-OUTPOSTS-BUCKET`* (il bucket di destinazione) permettono a S3 su Outposts di replicare gli oggetti o eliminare i contrassegni nel bucket Outposts di destinazione. Per informazioni sui contrassegni di eliminazione, consulta la sezione Effetto delle operazioni di eliminazione sulla replica.

  > **Note**
  >
  > - Le autorizzazioni per l'operazione `s3-outposts:ReplicateObject` nel bucket *`DESTINATION-OUTPOSTS-BUCKET`* (il bucket di destinazione) consentono anche la replica dei tag degli oggetti. Pertanto non è necessario concedere esplicitamente l'autorizzazione per l'operazione `s3-outposts:ReplicateTags`.
  > - Per la replica tra account, il proprietario del bucket Outposts di destinazione deve aggiornare la policy dei bucket per concedere l'autorizzazione per l'operazione `s3-outposts:ReplicateObject` nel *`DESTINATION-OUTPOSTS-BUCKET`*. L'operazione `s3-outposts:ReplicateObject` consente a S3 su Outposts di replicare oggetti e tag nel bucket Outposts di destinazione.

Per un elenco delle operazioni di S3 on Outposts, consulta Operazioni definite da Amazon S3 su Outposts.

> **Important**
>
> L'Account AWS che possiede il ruolo IAM deve disporre delle autorizzazioni per le operazioni che concede al ruolo.
> Supponi ad esempio che il bucket Outposts di origine contenga oggetti di proprietà di un altro Account AWS. Occorre che il proprietario degli oggetti conceda esplicitamente le autorizzazione richieste all'Account AWS proprietario del ruolo IAM tramite la policy dei bucket e la policy dei punti di accesso. In caso contrario, S3 su Outposts non può accedere agli oggetti e la replica degli oggetti ha esito negativo.
> Le autorizzazioni descritte si riferiscono alla configurazione di replica minima. Se scegli di aggiungere configurazioni di replica facoltative, devi concedere ulteriori autorizzazioni a S3 su Outposts.

Concessione di autorizzazioni quando i bucket Outposts di origine e di destinazione sono di proprietà di Account AWS diversi

Quando i bucket Outposts di origine e di destinazione non sono di proprietà degli stessi account, il proprietario del bucket Outposts di destinazione deve aggiornare le policy dei bucket e dei punti di accesso per il bucket di destinazione. Queste policy devono concedere al proprietario del bucket Outposts di origine e al ruolo del servizio IAM le autorizzazioni per eseguire le operazioni di replica, come mostrato nei seguenti esempi di policy. In caso contrario la replica avrà esito negativo. In questi esempi di policy, *`DESTINATION-OUTPOSTS-BUCKET`* è il bucket di destinazione. Per usare questi esempi di policy, sostituisci *`user input placeholders`* con le tue informazioni.

Se stai creando il ruolo di servizio IAM manualmente, imposta il percorso del ruolo come `role/service-role/`, nel modo mostrato nei seguenti esempi di policy. Per ulteriori informazioni, consulta ARN IAM nella Guida per l'utente di 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**
>
> In presenza di oggetti con tag nel bucket Outposts di origine, tenere in considerazione quanto segue:
> Se il proprietario del bucket Outposts di origine concede a S3 su Outposts l'autorizzazione per le operazioni `s3-outposts:GetObjectVersionTagging` e `s3-outposts:ReplicateTags` per replicare i tag degli oggetti (tramite il ruolo IAM), Amazon S3 replica i tag insieme agli oggetti. Per informazioni sul ruolo IAM, consulta Creazione di un ruolo IAM.

Creazione delle regole di replica su Outposts

La Replica S3 su Outposts è la copia asincrona e automatica degli oggetti di vari bucket negli stessi o in diversi AWS Outposts. Il processo replica gli oggetti appena creati e gli aggiornamenti degli oggetti da un bucket Outposts di origine in uno o più bucket Outposts di destinazione. Per ulteriori informazioni, consulta Replica degli oggetti per S3 su Outposts.

> **Note**
>
> Gli oggetti esistenti nel bucket Outposts di origine prima della configurazione delle regole di replica non vengono replicati. In altre parole, S3 su Outposts non esegue la replica degli oggetti in modo retroattivo. Per replicare gli oggetti creati prima della configurazione della replica, puoi utilizzare l'operazione API `CopyObject` per copiarli nello stesso bucket. Dopo la copia, gli oggetti vengono visualizzati come "nuovi" nel bucket e quindi viene loro applicata la configurazione della replica. Per ulteriori informazioni sulla copia di un oggetto, consulta Copia di un oggetto in un bucket Amazon S3 su Outposts utilizzando AWS SDK for Java e CopyObject nella Documentazione di riferimento delle API di Amazon Simple Storage Service.

Quando si configura la replica, vengono aggiunte le regole di replica al bucket Outposts di origine. Le regole di replica definiscono gli oggetti del bucket Outposts di origine da replicare e i bucket Outposts di destinazione in cui vengono archiviati gli oggetti replicati. È possibile creare una regola per replicare tutti gli oggetti in un bucket o un sottoinsieme di oggetti con un prefisso di nome di chiave specifico, uno o più tag di oggetto o entrambi gli elementi. Un bucket Outposts di destinazione può trovarsi nello stesso Outpost del bucket Outposts di origine o in un Outpost diverso.

Per le regole di replica di S3 su Outposts, devi fornire il nome della risorsa Amazon (ARN) del punto di accesso del bucket Outposts di origine e il nome della risorsa Amazon (ARN) del punto di accesso del bucket Outposts di destinazione anziché i nomi dei bucket Outposts di origine e di destinazione.

Se specifichi l'ID della versione dell'oggetto da eliminare, S3 su Outposts elimina la versione dell'oggetto nel bucket Outposts di origine. Tuttavia non replica l'eliminazione nel bucket Outposts di destinazione. In altre parole, non elimina la stessa versione dell'oggetto dal bucket Outposts di destinazione. Questo comportamento permette di proteggere i dati da eliminazioni da parte di utenti malintenzionati.

Quando si aggiunge una regola di replica a un bucket Outposts, la regola viene abilitata per impostazione predefinita e pertanto inizia a funzionare non appena viene salvata.

In questo esempio viene configurata la replica per i bucket Outposts di origine e di destinazione in Outpost diversi e di proprietà dello stesso Account AWS. Sono forniti esempi per utilizzare la console di Amazon S3, la AWS Command Line Interface (AWS CLI) e AWS SDK for Java e AWS SDK for .NET. Per ulteriori informazioni sulle autorizzazioni della Replica S3 su Outposts tra account, consulta Concessione di autorizzazioni quando i bucket Outposts di origine e di destinazione sono di proprietà di Account AWS diversi.

Per i prerequisiti per configurare le regole di replica di S3 su Outposts, consulta Prerequisiti per la creazione delle regole di replica.

Utilizzo della console S3

Segui questi passaggi per configurare una regola di replica quando il bucket Amazon S3 su Outposts di destinazione si trova in un Outpost diverso rispetto al bucket Outposts di origine.

Se il bucket Outposts di destinazione si trova in un account diverso rispetto al bucket Outposts di origine, è necessario aggiungere al bucket Outposts di destinazione una policy dei bucket per fornire al proprietario dell'account del bucket Outposts di origine l'autorizzazione per replicare gli oggetti nel bucket Outposts di destinazione. Per ulteriori informazioni, consulta Concessione delle autorizzazioni quando i bucket di origine e di destinazione sono di proprietà di diversi Account AWS.

Per creare una regola di replica

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Bucket Outposts scegli il nome del bucket che desideri utilizzare come bucket di origine.

3. Seleziona la scheda Gestione, scorri verso il basso fino a Regole di replica e quindi scegli Crea regola di replica.
4. In Nome della regola di replica specifica un nome per la regola al fine di poterla identificare in un secondo momento. Il nome è obbligatorio e deve essere univoco all'interno del bucket.
5. In Stato, l'opzione Abilitato è selezionata per impostazione predefinita. Una regola abilitata inizia a funzionare non appena viene salvata. Se desideri abilitare la regola in un secondo momento, seleziona Disabilitata.
6. In Priorità, il valore di priorità della regola determina quale regola viene applicata in caso di sovrapposizione delle regole. Quando gli oggetti sono inclusi nell'ambito di più regole di replica, S3 su Outposts utilizza questi valori di priorità per evitare i conflitti. Per impostazione predefinita, le nuove regole vengono aggiunte alla configurazione di replica con la priorità più alta. Più elevato è il numero, maggiore è la priorità.

   Per modificare la priorità della regola, dopo averla salvata, scegli il nome della regola dall'elenco delle regole di replica, seleziona Operazioni e quindi scegli Modifica priorità.
7. In Bucket di origine sono disponibili le seguenti opzioni per l'impostazione dell'origine della replica:

   - Per replicare l'intero bucket, scegli Applica a tutti gli oggetti nel bucket.
   - Per applicare il filtro di prefisso o tag all'origine di replica, scegli Limita l'ambito della regola utilizzando uno o più filtri. È possibile combinare un prefisso con i tag.
     - Per replicare tutti gli oggetti con lo stesso prefisso, immetti un prefisso nella casella Prefisso. Con il filtro Prefisso limita la replica a tutti gli oggetti con nomi che iniziano con la stessa stringa (ad esempio, `pictures`).

       Se immetti un prefisso corrispondente al nome di una cartella, devi utilizzare una / (barra) come ultimo carattere (ad esempio, `pictures/`).
     - Per replicare tutti gli oggetti contenenti uno o più tag di oggetto, seleziona Aggiungi tag e specifica la coppia valore-chiave nelle caselle. Per aggiungere un altro tag, ripeti la procedura. Per ulteriori informazioni sui tag degli oggetti, consulta Aggiunta di tag per bucket S3 su Outposts.
8. Per accedere al bucket di origine S3 su Outposts per la replica, in Nome del punto di accesso di origine, scegli un punto di accesso associato al bucket di origine.
9. In Destinazione, scegli il nome della risorsa Amazon (ARN) del punto di accesso del bucket Outposts di destinazione in cui desideri che S3 su Outposts replichi gli oggetti. I bucket Outposts

di destinazione possono trovarsi nello stesso o in diversi Account AWS del bucket Outposts di origine.

Se il bucket di destinazione si trova in un account diverso rispetto al bucket Outposts di origine, è necessario aggiungere al bucket Outposts di destinazione una policy dei bucket per fornire al proprietario dell'account del bucket Outposts di origine l'autorizzazione per replicare gli oggetti nel bucket Outposts di destinazione. Per ulteriori informazioni, consulta Concessione di autorizzazioni quando i bucket Outposts di origine e di destinazione sono di proprietà di Account AWS diversi.

> **Note**
>
> Se la funzione Controllo delle versioni non è abilitata nel bucket Outposts di destinazione, viene visualizzato un messaggio di avviso contenente un pulsante Abilita Controllo delle versioni. Seleziona questo pulsante per abilitare la funzione Controllo delle versioni nel bucket.

10. Configura un ruolo di servizio AWS Identity and Access Management (IAM) che S3 su Outposts può assumere per replicare gli oggetti per tuo conto.

    Per impostare un ruolo IAM, in Ruolo IAM effettua una delle operazioni seguenti:

    - Per fare in modo che S3 su Outposts crei un nuovo ruolo IAM per la configurazione di replica, seleziona Scegli tra i ruoli IAM esistenti e quindi Crea nuovo ruolo. Quando salvi la regola, viene generata una nuova policy per il ruolo IAM corrispondente ai bucket Outposts di origine e di destinazione scelti. Consigliamo di scegliere Crea nuovo ruolo.
    - Puoi anche decidere di utilizzare un ruolo IAM esistente. In tal caso, è necessario scegliere un ruolo che conceda a S3 su Outposts le autorizzazioni necessarie per la replica. Se questo ruolo non concede autorizzazioni sufficienti a S3 su Outposts per seguire la regola di replica, la replica non riesce.

      Per scegliere un ruolo esistente, seleziona Scegli tra i ruoli IAM esistenti e scegli il ruolo nel menu a discesa. Puoi anche scegliere Inserisci l'ARN del ruolo IAM e quindi inserire il nome della risorsa Amazon (ARN) del ruolo IAM.

> **Important**
>
> Quando aggiungi una regola di replica a un bucket S3 su Outposts, è necessario disporre delle autorizzazioni `iam:CreateRole` e `iam:PassRole` per poter creare e trasferire il ruolo IAM che concede le autorizzazioni di replica a S3 su Outposts. Per ulteriori informazioni, consulta Concessione di autorizzazioni utente per il passaggio di un ruolo a un Servizio AWS nella Guida per l'utente di IAM.

11. Tutti gli oggetti nei bucket Outposts sono crittografati per impostazione predefinita. Per ulteriori informazioni sulla crittografia di S3 su Outposts, consulta Crittografia dei dati in S3 su Outposts. È possibile replicare solo gli oggetti crittografati utilizzando la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3). La replica degli oggetti crittografati utilizzando la crittografia lato server con le chiavi AWS Key Management Service (AWS KMS) (SSE-KMS) o la crittografia lato server con le chiavi di crittografia fornite dal cliente (SSE-C) non è supportata.
12. Se necessario, abilita le seguenti opzioni aggiuntive durante l'impostazione della configurazione della regola di replica:
    - Se desideri abilitare i parametri della replica S3 su Outposts nella configurazione di replica, seleziona Parametri di replica. Per ulteriori informazioni, consulta Monitoraggio dell'avanzamento con le metriche relative alla replica.
    - Se desideri abilitare la replica del contrassegno di eliminazione nella configurazione di replica, seleziona Replica del contrassegno di eliminazione. Per ulteriori informazioni, consulta Effetto delle operazioni di eliminazione sulla replica.
    - Se desideri replicare le modifiche ai metadati apportate alle repliche negli oggetti di origine, seleziona Sincronizzazione delle modifiche delle repliche. Per ulteriori informazioni, consulta Stato della replica se su Outposts è abilitata la sincronizzazione della modifica della replica Amazon S3.
13. Per finire, scegli Crea regola.

Dopo aver salvato la regola, è possibile modificarla, abilitarla, disabilitarla o eliminarla. Per eseguire queste operazioni, vai alla scheda Gestione del bucket Outposts di origine, scorri verso il basso fino alla sezione Regole di replica, scegli la tua regola e quindi scegli Modifica regola.

Utilizzo di AWS CLI

Per utilizzare la AWS CLI per impostare la replica quando i bucket Outposts di origine e di destinazione sono di proprietà dello stesso Account AWS, esegui le seguenti operazioni:

- Crea i bucket Outposts di origine e di destinazione.
- Abilita il controllo delle versioni su entrambi i bucket.
- Crea un ruolo IAM che permette a S3 su Outposts di replicare gli oggetti.
- Aggiungi la configurazione di replica al bucket Outposts di origine.

Per verificare l'impostazione, testarla.

Per impostare la replica quando i bucket Outposts di origine e di destinazione sono di proprietà dello stesso Account AWS

1. Impostare un profilo di credenziali per la AWS CLI. utilizzando il nome profilo `acctA`. Per informazioni sull'impostazione di profili con credenziali, consulta Profili denominati nella Guida per l'utente di AWS Command Line Interface.

   > **Important**
   >
   > Il profilo utilizzato per questo esercizio deve disporre delle autorizzazioni necessarie. Ad esempio, nella configurazione di replica dovrai specificare il ruolo di servizio IAM che S3 su Outposts può assumere. Puoi effettuare questa operazione solo se il profilo che utilizzi dispone delle autorizzazioni `iam:CreateRole` e `iam:PassRole`. Per ulteriori informazioni, consulta Concessione di autorizzazioni utente per il passaggio di un ruolo a un Servizio AWS nella Guida per l'utente di IAM. Se utilizzi le credenziali di amministratore per creare un profilo denominato, il profilo denominato avrà le autorizzazioni necessarie per eseguire tutte le attività.

2. Creare un bucket di origine e abilitare la funzione Controllo delle versioni. Il comando `create-bucket` seguente crea un bucket *`SOURCE-OUTPOSTS-BUCKET`* nella regione Stati Uniti orientali (Virginia settentrionale) (`us-east-1`). Per utilizzare questo comando, sostituisci *`user input placeholders`* con le tue specifiche informazioni.

   ```
   aws s3control create-bucket --bucket SOURCE-OUTPOSTS-BUCKET --outpost-id SOURCE-OUTPOST-ID --profile acctA --region us-east-1
   ```

Il comando `put-bucket-versioning` seguente abilita il controllo delle versioni sul bucket *SOURCE-OUTPOSTS-BUCKET*. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue specifiche informazioni.

```
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. Creare un bucket di destinazione e abilitare la funzione Controllo delle versioni. Il comando `create-bucket` seguente crea un bucket *DESTINATION-OUTPOSTS-BUCKET* nella regione Stati Uniti occidentali (Oregon) (`us-west-2`). Per utilizzare questo comando, sostituisci *user input placeholders* con le tue specifiche informazioni.

> **Note**
>
> Per impostare la configurazione di replica quando entrambi i bucket Outposts di origine e di destinazione si trovano nello stesso Account AWS si utilizza lo stesso profilo denominato. Questo esempio usa `acctA`. Per testare la configurazione della replica quando i bucket sono di proprietà di Account AWS diversi, occorre specificare profili differenti per ciascun 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
```

Il comando `put-bucket-versioning` seguente abilita il controllo delle versioni sul bucket *DESTINATION-OUTPOSTS-BUCKET*. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue specifiche informazioni.

```
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. Crea un ruolo di servizio IAM. Aggiungi questo ruolo di servizio al bucket *SOURCE-OUTPOSTS-BUCKET* in un secondo momento nella configurazione di replica. S3 su Outposts assume questo ruolo per replicare gli oggetti per tuo conto. Il ruolo IAM si crea in due fasi:

a. Crea un ruolo IAM.

   i. Copiare la seguente policy di attendibilità e salvarla in un file denominato `s3-on-outposts-role-trust-policy.json` nella directory corrente sul computer locale. Questa policy fornisce a S3 su Outposts le autorizzazioni ai principali del servizio per assumere il ruolo di servizio.

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

   ii. Per creare un ruolo, eseguire il comando seguente. Sostituire *`user input placeholders`* con le proprie informazioni.

```
aws iam create-role --role-name replicationRole --assume-role-policy-
document file://s3-on-outposts-role-trust-policy.json --profile acctA
```

b. Collega una policy di autorizzazioni al ruolo di servizio.

   i. Copiare la seguente policy di autorizzazioni e salvarla in un file denominato `s3-on-outposts-role-permissions-policy.json` nella directory corrente sul computer locale. Questa policy fornisce le autorizzazioni per varie operazioni su oggetti e bucket S3 su Outposts. Per utilizzare questa policy, sostituisci *`user input placeholders`* con le tue specifiche informazioni.

```
{
   "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. Eseguire il comando seguente per creare una policy e collegarla al ruolo. Sostituire *user input placeholders* con le proprie informazioni.

```
aws iam put-role-policy --role-name replicationRole --policy-
document file://s3-on-outposts-role-permissions-policy.json --policy-
name replicationRolePolicy --profile acctA
```

5. Aggiungi la configurazione di replica al bucket *SOURCE-OUTPOSTS-BUCKET*.

   a. Anche se l'API S3 su Outposts richiede la configurazione di replica in formato XML, la AWS CLI richiede di specificare la configurazione di replica in formato JSON. Salvare il seguente JSON in un file denominato `replication.json` nella directory locale sul computer in uso. Per utilizzare questa configurazione, sostituisci *user input placeholders* con le tue specifiche informazioni.

```
{
  "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. Esegui il comando `put-bucket-replication` seguente per aggiungere la configurazione di replica al bucket Outposts di origine. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue specifiche informazioni.

```
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. Per recuperare la configurazione di replica, utilizzare il comando `get-bucket-replication`. Per utilizzare questo comando, sostituisci *user input placeholders* con le tue specifiche informazioni.

```
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. Verifica la configurazione nella console di Amazon S3:

a. Accedi alla AWS Management Console e apri la console di Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

b. Nel bucket di *SOURCE-OUTPOSTS-BUCKET*, crea una cartella denominata `Tax`.

c. Aggiungi oggetti di esempio alla cartella `Tax` del bucket di *SOURCE-OUTPOSTS-BUCKET*.

d. Nel bucket di *DESTINATION-OUTPOSTS-BUCKET*, verifica quanto segue:

- S3 su Outposts ha replicato gli oggetti.

> **Note**
>
> Il tempo richiesto da S3 su Outposts per la replica di un oggetto dipende dalle dimensioni dell'oggetto. Per informazioni su come visualizzare lo stato della replica, consulta la sezione Ottenimento delle informazioni sullo stato della replica.

- Nella scheda Proprietà dell'oggetto, lo Stato di replica è impostato su Replica (che identifica l'oggetto come replica).

Gestione della replica

In questa sezione vengono descritte ulteriori opzioni per la configurazione della replica disponibili in S3 su Outposts, nonché viene spiegato come determinare lo stato della replica e come risolvere i problemi della replica. Per informazioni sulla configurazione della replica di base, consulta Impostazione della replica.

Argomenti

- Monitoraggio dell'avanzamento con le metriche relative alla replica
- Ottenimento delle informazioni sullo stato della replica
- Risoluzione dei problemi nella replica
- Utilizzo di EventBridge per la replica S3 su Outposts

Monitoraggio dell'avanzamento con le metriche relative alla replica

Replica Amazon S3 su Outposts fornisce metriche dettagliate per le regole di replica nella configurazione della replica. Con le metriche relative alla replica puoi monitorare l'avanzamento della replica a intervalli di 5 minuti tramite il tracciamento dei byte in attesa di replica, della latenza della replica e delle operazioni in attesa di replica. A supporto della procedura di risoluzione dei problemi relativi alla configurazione, puoi anche configurare Amazon EventBridge in modo da ricevere notifiche relative agli errore di replica.

Se le metriche di replica sono abilitate, Replica Amazon S3 su Outposts pubblicano le seguenti metriche su Amazon CloudWatch:

- Byte in attesa di replica: il numero totale di byte di oggetti in attesa di replica per una determinata regola di replica.

- Latenza di replica: il numero massimo di secondi entro i quali i bucket di destinazione della replica sono in ritardo rispetto al bucket di origine per una determinata regola di replica.
- Operazioni in attesa di replica: il numero di operazioni in attesa di replica per una determinata regola di replica. Le operazioni includono oggetti, contrassegni di eliminazione e tag.

> **Note**
>
> Le metriche di Replica Amazon S3 su Outposts vengono fatturate alla stessa tariffa delle metriche personalizzate di Amazon CloudWatch. Per ulteriori informazioni, consultare la pagina dei prezzi di CloudWatch

Ottenimento delle informazioni sullo stato della replica

Lo stato della replica consente di determinare lo stato corrente di un oggetto sottoposto a replica in Amazon S3 su Outposts. Lo stato della replica di un oggetto di origine restituirà `PENDING`, `COMPLETED` o `FAILED`. Lo stato della replica di una replica restituirà `REPLICA`.

Panoramica dello stato della replica

In uno scenario di replica, esistono un bucket di origine in cui si configura la replica e un bucket di destinazione in cui S3 su Outposts replica gli oggetti. Quando richiedi un oggetto (tramite `GetObject`) o i metadati di un oggetto (tramite `HeadObject`) da questi bucket, S3 su Outposts restituisce l'intestazione `x-amz-replication-status` nella risposta come segue:

- Quando richiedi un oggetto dal bucket di origine, S3 su Outposts restituisce l'intestazione `x-amz-replication-status` se l'oggetto nella richiesta è idoneo per la replica.

  Supponiamo, ad esempio, che nella configurazione della replica venga specificato il prefisso di oggetto `TaxDocs` che indica a S3 su Outposts di replicare solo gli oggetti con il prefisso del nome della chiave `TaxDocs`. Tutti gli oggetti caricati che hanno questo prefisso del nome della chiave, ad esempio `TaxDocs/document1.pdf`, verranno replicati. Per le richieste di oggetti con questo prefisso del nome della chiave, S3 su Outposts restituisce l'intestazione `x-amz-replication-status` con uno dei valori seguenti per lo stato della replica dell'oggetto: `PENDING`, `COMPLETED` o `FAILED`.

> **Note**
>
> Se la replica dell'oggetto ha esito negativo dopo il caricamento di un oggetto, non è possibile provare a eseguirla di nuovo. È necessario caricare di nuovo l'oggetto. Gli oggetti passano a uno stato `FAILED` per problemi dovuti ad esempio alla mancanza di autorizzazioni per il ruolo di replica o autorizzazioni di bucket mancanti. In caso di errori temporanei, ad esempio se un bucket o un outpost non è disponibile, lo stato della replica non passerà a `FAILED`, ma rimarrà `PENDING`. Dopo che la risorsa è tornata online, S3 su Outposts riprenderà la replica di tali oggetti.

- Quando richiedi un oggetto da un bucket di destinazione, se l'oggetto nella richiesta è una replica creata da S3 su Outposts, S3 su Outposts restituisce l'intestazione `x-amz-replication-status` con il valore `REPLICA`.

> **Note**
>
> Prima di eliminare un oggetto da un bucket di origine in cui è abilitata la replica, è consigliabile controllare lo stato della replica per assicurarsi che l'oggetto sia stato replicato.

Stato della replica se su Outposts è abilitata la sincronizzazione della modifica della replica Amazon S3

Quando le regole di replica abilitano la sincronizzazione delle modifiche della replica S3 su Outposts, le repliche possono riportare stati diversi da `REPLICA`. Se le modifiche dei metadati sono in corso di replica, l'intestazione `x-amz-replication-status` della replica restituisce `PENDING`. Se la sincronizzazione delle modifiche della replica non riesce a replicare i metadati, l'intestazione della replica restituisce `FAILED`. Se i metadati vengono replicati correttamente, l'intestazione della replica restituisce il valore `REPLICA`.

Risoluzione dei problemi nella replica

Se le repliche degli oggetti non vengono visualizzate nel bucket Amazon S3 su Outposts di destinazione dopo aver configurato la replica, usa questi suggerimenti per identificare e risolvere i problemi.

- Il tempo impiegato da S3 su Outposts per replicare un oggetto dipende da diversi fattori, tra cui la distanza tra gli outpost di origine e destinazione e le dimensioni dell'oggetto.

  È possibile controllare lo stato della replica dell'oggetto di origine. Se lo stato della replica dell'oggetto è `PENDING`, significa che S3 su Outposts non ha completato la replica. Se lo stato della replica dell'oggetto è `FAILED`, controlla la configurazione della replica impostata nel bucket di origine.

- Nella configurazione di replica nel bucket di origine verifica quanto segue:
  - La correttezza del nome della risorsa Amazon (ARN) del punto di accesso relativo al bucket di destinazione.
  - La correttezza del prefisso del nome della chiave. Ad esempio, se si imposta la configurazione per replicare gli oggetti con il prefisso `Tax`, solo gli oggetti con i nomi della chiave quali `Tax/document1` o `Tax/document2` vengono replicati. Un oggetto con il nome della chiave `document3` non sia replicato.
  - Che lo stato sia `Enabled`.
- Verifica che il controllo delle versioni non sia stato sospeso per nessuno dei bucket. Sia per il bucket di origine che per quello di destinazione deve essere abilitata la funzione Controllo delle versioni.
- Se il bucket di destinazione è di proprietà di un altro Account AWS, verifica che il proprietario di tale bucket disponga di una policy del bucket che consenta al proprietario del bucket di origine di replicare gli oggetti. Per un esempio, consulta Concessione di autorizzazioni quando i bucket Outposts di origine e di destinazione sono di proprietà di Account AWS diversi.
- Se la replica di un oggetto non è presente nel bucket di destinazione, il problema potrebbe essere dovuto alle cause seguenti:
  - S3 su Outposts non replica un oggetto in un bucket di origine che è una replica creata da un'altra configurazione della replica. Se, ad esempio, imposti una configurazione della replica dal bucket A al bucket B al bucket C, S3 su Outposts non replica le repliche degli oggetti del bucket B nel bucket C.

    Se desideri replicare gli oggetti del bucket A nel bucket B e nel bucket C, imposta più destinazioni di bucket in regole di replica diverse per la configurazione della replica del bucket di origine. Ad esempio, crea due regole di replica sul bucket di origine A, con una regola da replicare nel bucket di destinazione B e l'altra regola da replicare nel bucket di destinazione C.
  - Il proprietario di un bucket di origine può concedere ad altri Account AWS l'autorizzazione necessaria per caricare gli oggetti. Per impostazione predefinita, il proprietario del bucket di

origine non dispone di autorizzazioni per gli oggetti creati da altri account. La configurazione di replica esegue la replica solo degli oggetti per i quali il proprietario del bucket di origine dispone delle autorizzazioni di accesso. Per evitare problemi di replica, il proprietario del bucket di origine può concedere ad altri Account AWS le autorizzazioni necessarie per creare oggetti in modo condizionale, richiedendo autorizzazioni di accesso esplicite su quegli oggetti. Per un esempio di policy, consulta Concedere autorizzazioni multi-account per il caricamento di oggetti a garanzia del controllo completo da parte del proprietario del bucket.

- Supponiamo di aggiungere nella configurazione della replica una regola per replicare un sottoinsieme di oggetti con un tag specifico. In questo caso, è necessario assegnare il valore e la chiave del tag specifici al momento della creazione dell'oggetto per permettere a S3 su Outposts di replicare l'oggetto. Se prima crei un oggetto e quindi aggiungi il tag a tale oggetto, S3 su Outposts non replica l'oggetto.
- La replica non riesce se la policy del bucket nega l'accesso al ruolo di replica per una delle seguenti operazioni:

  Bucket di origine:

  ```
  "s3-outposts:GetObjectVersionForReplication",
  "s3-outposts:GetObjectVersionTagging"
  ```

  Bucket di destinazione:

  ```
  "s3-outposts:ReplicateObject",
  "s3-outposts:ReplicateDelete",
  "s3-outposts:ReplicateTags"
  ```

- Amazon EventBridge può avvisarti quando gli oggetti non vengono replicati negli outpost di destinazione. Per ulteriori informazioni, consulta Utilizzo di EventBridge per la replica S3 su Outposts.

Utilizzo di EventBridge per la replica S3 su Outposts

Amazon S3 su Outposts è integrato con Amazon EventBridge e utilizza lo spazio dei nomi `s3-outposts`. EventBridge è un servizio routing di eventi serverless che puoi utilizzare per connettere le applicazioni ai dati provenienti da un'ampia gamma di origini. Per ulteriori informazioni, consulta Che cos'è Amazon EventBridge? nella Guida per l'utente di Amazon EventBridge.

A supporto delle procedure di risoluzione dei problemi relativi alla configurazione della replica, puoi configurare Amazon EventBridge in modo da ricevere le notifiche relative agli eventi di errore relativi alla replica. EventBridge può inviare una notifica quando gli oggetti non vengono replicati nell'istanza Outposts di destinazione. Per ulteriori informazioni sullo stato corrente di un oggetto da replicare, consulta Panoramica dello stato della replica.

Amazon S3 può inviare eventi a EventBridge ogni volta che determinati eventi si verificano nel bucket Outposts. A differenza di altre destinazioni, non è necessario selezionare i tipi di eventi che si desidera inviare. È anche possibile utilizzare le regole di EventBridge per instradare gli eventi a destinazioni aggiuntive. Dopo aver abilitato EventBridge, S3 su Outposts invia tutti i seguenti eventi a EventBridge.

Tipo di evento	Descrizione	Spazio dei nomi
`Operation FailedReplication`	La replica di un oggetto all'interno di una regola di replica non è riuscita. Per ulteriori informazioni sui motivi degli errori della replica S3 su Outposts, consulta Utilizzo di EventBridge per visualizzare i motivi degli errori della replica S3 su Outposts.	`s3-outposts`

Utilizzo di EventBridge per visualizzare i motivi degli errori della replica S3 su Outposts

La seguente tabella elenca i motivi degli errori di replica in S3 su Outposts. È possibile configurare una regola di EventBridge per pubblicare e visualizzare il motivo dell'errore tramite Amazon Simple Queue Service (Amazon SQS), Servizio di notifica semplice Amazon (Amazon SNS), AWS Lambda o File di log Amazon CloudWatch. Per ulteriori informazioni sulle autorizzazioni richieste per utilizzare queste risorse per EventBridge, consulta l'argomento relativo all'uso delle policy basate su risorse per Amazon EventBridge.

Motivo dell'errore di replica	Descrizione
`AssumeRoleNotPermitted`	S3 su Outposts non può assumere il ruolo AWS Identity and Access Management (IAM) specificato nella configurazione della replica.

Motivo dell'errore di replica	Descrizione
`DstBucketNotFound`	S3 su Outposts non è in grado di trovare il bucket di destinazione specificato nella configurazione della replica.
`DstBucketUnversioned`	Il controllo delle versioni non è abilitato nel bucket Outposts di destinazione. Per replicare gli oggetti con Replica Amazon S3 su Outposts, devi abilitare il controllo delle versioni nel bucket di destinazione.
`DstDelObjNotPermitted`	S3 su Outposts non è in grado di replicare le eliminazioni nel bucket di destinazione. È possibile che manchi l'autorizzazione `s3-outposts:ReplicateDelete` per il bucket di destinazione.
`DstMultipartCompleteNotPermitted`	S3 su Outposts non è in grado di completare il caricamento degli oggetti in più parti nel bucket di destinazione. È possibile che manchi l'autorizzazione `s3-outposts:ReplicateObject` per il bucket di destinazione.
`DstMultipartInitNotPermitted`	S3 su Outposts non è in grado di avviare il caricamento degli oggetti in più parti nel bucket di destinazione. È possibile che manchi l'autorizzazione `s3-outposts:ReplicateObject` per il bucket di destinazione.
`DstMultipartPartUploadNotPermitted`	S3 su Outposts non è in grado di eseguire il caricamento degli oggetti in più parti nel bucket di destinazione. È possibile che manchi l'autorizzazione `s3-outposts:ReplicateObject` per il bucket di destinazione.

Motivo dell'errore di replica	Descrizione
`DstOutOfCapacity`	S3 su Outposts non è in grado di eseguire la replica nell'outpost di destinazione perché è stata esaurita la capacità di archiviazione S3.
`DstPutObjNotPermitted`	S3 su Outposts non è in grado di replicare gli oggetti nel bucket di destinazione. È possibile che manchi l'autorizzazione `s3-outposts:ReplicateObject` per il bucket di destinazione.
`DstPutTaggingNotPermitted`	S3 su Outposts non è in grado di replicare tag di oggetti nel bucket di destinazione. È possibile che manchi l'autorizzazione `s3-outposts:ReplicateObject` per il bucket di destinazione.
`DstVersionNotFound`	S3 su Outposts non è in grado di trovare la versione dell'oggetto richiesta nel bucket di destinazione per replicare i metadati di tale versione.
`SrcBucketReplicationConfigMissing`	S3 su Outposts non è in grado di trovare una configurazione della replica per il punto di accesso associato al bucket Outposts di origine.
`SrcGetObjNotPermitted`	S3 su Outposts non è in grado di accedere all'oggetto nel bucket di origine per la replica. È possibile che manchi l'autorizzazione `s3-outposts:GetObjectVersionForReplication` per il bucket di origine.

Motivo dell'errore di replica	Descrizione
`SrcGetTaggingNotPermitted`	S3 su Outposts non è in grado di accedere alle informazioni sui tag degli oggetti dal bucket di origine. È possibile che manchi l'autorizzazione `s3-outposts:GetObjectVersionTagging` per il bucket di origine.
`SrcHeadObjectNotPermitted`	S3 su Outposts non è in grado di recuperare i metadati degli oggetti dal bucket di origine. È possibile che manchi l'autorizzazione `s3-outposts:GetObjectVersionForReplication` per il bucket di origine.
`SrcObjectNotEligible`	L'oggetto non è idoneo per la replica. L'oggetto o i relativi tag non corrispondono alla configurazione della replica.

Per ulteriori informazioni sulla risoluzione dei problemi relativi alla replica, consulta i seguenti argomenti:

- Creazione di un ruolo IAM
- Risoluzione dei problemi nella replica

Monitoraggio di EventBridge con CloudWatch

Per il monitoraggio Amazon EventBridge è integrato con Amazon CloudWatch. EventBridge invia automaticamente le metriche a CloudWatch a intervalli di un minuto. Queste metriche includono il numero di eventi abbinati da una regola e il numero di volte in cui una destinazione viene richiamata da una regola. Quando una regola viene eseguita in EventBridge, vengono richiamati tutte le destinazioni associate alla regola. È possibile monitorare il comportamento di EventBridge tramite CloudWatch nei seguenti modi.

- Puoi monitorare le metriche EventBridge disponibili per le regole EventBridge dal pannello di controllo di CloudWatch. Puoi quindi utilizzare le funzionalità di CloudWatch, come gli allarmi

CloudWatch, per impostare allarmi per determinate metriche. Se tali metriche raggiungono i valori di soglia personalizzati specificati negli allarmi, riceverai notifiche e potrai agire di conseguenza.

- Puoi impostare File di log Amazon CloudWatch come destinazione della regola EventBridge. EventBridge crea quindi flussi di log e File di log Amazon CloudWatch memorizza il testo degli eventi come voci di log. Per ulteriori informazioni, consulta l'argomento relativo a EventBridge e File di log Amazon CloudWatch.

Per ulteriori informazioni sul debug della consegna e sull'archiviazione degli eventi EventBridge, consulta i seguenti argomenti:

- Policy di ripetizione degli eventi e utilizzo delle code DLQ
- Archiviazione degli eventi EventBridge

Condivisione di S3 su Outposts utilizzando AWS RAM

Amazon S3 on Outposts supporta la condivisione della capacità S3 tra più account all'interno di un'organizzazione utilizzando (). AWS Resource Access Manager AWS RAM Con la condivisione di S3 on Outposts, puoi consentire ad altri di creare e gestire bucket, endpoint e punti di accesso sul tuo Outpost.

Questo argomento dimostra come utilizzare per AWS RAM condividere S3 on Outposts e le risorse correlate con altri Account AWS membri dell'organizzazione. AWS

Prerequisiti

- L'account proprietario dell'Outpost ha un'organizzazione configurata in AWS Organizations. Per ulteriori informazioni sulla configurazione di un'organizzazione, consulta Creazione di un'organizzazione nella Guida per l'utente di AWS Organizations .
- L'organizzazione include le persone con Account AWS cui desideri condividere la tua capacità S3 on Outposts. Per ulteriori informazioni, consulta la sezione Invio degli inviti agli Account AWS nella Guida per l'utente di AWS Organizations .
- Seleziona una delle seguenti opzioni per la condivisione. La seconda risorsa (Sottoreti o Outpost) deve essere selezionata in modo che siano accessibili anche gli endpoint. Gli endpoint sono un requisito di rete per accedere ai dati archiviati in S3 on Outposts.

Opzione 1	Opzione 2
S3 on Outposts Consente all'utente di creare bucket sugli Outpost e sui punti di accesso e di aggiungere oggetti a tali bucket. Sottoreti Consente all'utente di utilizzare il cloud privato virtuale (VPC) e gli endpoint associati alla sottorete.	S3 on Outposts Consente all'utente di creare bucket sugli Outpost e sui punti di accesso e di aggiungere oggetti a tali bucket. Outposts Consente all'utente di visualizzare i grafici di capacità S3 e la pagina iniziale della console AWS Outposts . Consente inoltre agli utenti di creare sottoreti su Outpost condivisi e di creare endpoint.

Procedura

1. Accedi a Outpost AWS Management Console utilizzando il proprietario Account AWS di Outpost, quindi apri la AWS RAM console all'indirizzo https://console.aws.amazon.com/ram.
2. Assicurati di aver abilitato la condivisione con AWS Organizations in AWS RAM. Per informazioni, consulta la sezione Abilitazione della condivisione con AWS Organizations nella Guida per l'utente di AWS RAM .
3. Utilizzare l'opzione 1 o l'opzione 2 nei prerequisiti per creare una condivisione di risorse. Se hai diverse risorse S3 su Outposts, seleziona gli Amazon Resource Name (ARN) delle risorse che desideri condividere. Per abilitare gli endpoint, condividi la sottorete o l'Outpost.

   Per ulteriori informazioni sulla creazione di una condivisione di risorse, consulta la sezione Creazione di una condivisione di risorse nella Guida per l'utente di AWS RAM .
4. La Account AWS persona con cui hai condiviso le tue risorse dovrebbe ora essere in grado di usare S3 su Outposts. A seconda dell'opzione selezionata nei prerequisiti, fornisci le seguenti informazioni all'utente dell'account:

Opzione 1	Opzione 2
L'ID dell'Outpost	L'ID dell'Outpost
L'ID del VPC	
L'ID sottorete	
L'ID del gruppo di sicurezza	

> **Note**
>
> L'utente può confermare che le risorse sono state condivise con lui utilizzando la AWS RAM console, AWS Command Line Interface (AWS CLI), gli AWS SDK o l'API REST. L'utente può visualizzare le proprie condivisioni di risorse esistenti utilizzando il get-resource-sharescomando CLI.

Esempi di utilizzo

Dopo aver condiviso le risorse S3 on Outposts con un altro account, tale account può gestire bucket e oggetti sul tuo Outpost. Se hai condiviso la risorsa Subnets (Sottoreti), tale account può utilizzare l'endpoint creato. Gli esempi seguenti mostrano come un utente può utilizzare Outpost AWS CLI per interagire con Outpost dopo aver condiviso queste risorse.

Example : creazione di un bucket

L'esempio seguente crea un bucket denominato *example-s3-bucket1* in Outpost. *op-01ac5d28a6a232904* Prima di utilizzare questo comando, sostituisci ciascun *user input placeholder* con i valori appropriati per il tuo caso d'uso.

```
aws s3control create-bucket --bucket example-s3-bucket1 --outpost-
id op-01ac5d28a6a232904
```

Per ulteriori informazioni su questo comando, consulta create-bucket nella Guida di riferimento AWS CLI .

Example : creazione di un punto di accesso

Nell'esempio seguente viene creato un punto di accesso su un Outpost utilizzando i parametri di esempio nella tabella seguente. Prima di utilizzare questo comando, sostituisci questi *user input placeholder* valori e il Regione AWS codice con i valori appropriati per il tuo caso d'uso.

Parameter	Valore
ID account	*111122223333*
Nome del punto di accesso	*example-outpost-access-point*
ID Outpost	*op-01ac5d28a6a232904*
Nome del bucket dell'Outpost	*example-s3-bucket1*
ID VPC	*vpc-1a2b3c4d5e6f7g8h9*

> **Note**
>
> Il parametro Account ID deve essere l' Account AWS ID del proprietario del bucket, che è l'utente condiviso.

```
aws s3control create-access-point --account-id 111122223333 --name example-outpost-
access-point \
--bucket arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/
bucket/example-s3-bucket1 \
--vpc-configuration VpcId=vpc-1a2b3c4d5e6f7g8h9
```

Per ulteriori informazioni su questo comando, vedere create-access-pointnella Guida di AWS CLI riferimento.

Example : caricamento di un oggetto

Nell'esempio seguente viene caricato il file *my_image.jpg* dal file system locale dell'utente in un oggetto denominato *images/my_image.jpg* tramite il punto di accesso *example-outpost-access-point* sull'Outpost *op-01ac5d28a6a232904*, di proprietà dell'account AWS

111122223333. Prima di utilizzare questo comando, sostituite questi *user input placeholder* valori e il Regione AWS codice con i valori appropriati per il vostro caso d'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
```

Per ulteriori informazioni, su questo comando, consulta put-object nella Guida di riferimento di AWS CLI .

> **Note**
>
> Se questa operazione si traduce in un errore Resource not found (Risorsa non trovata) o non risponde, il tuo VPC potrebbe non disporre di un endpoint condiviso.
> Per verificare se esiste un endpoint condiviso, usa il list-shared-endpoints AWS CLI comando. Se non esiste un endpoint condiviso, collabora con il proprietario di Outpost per crearne uno. Per ulteriori informazioni, consulta il riferimento ListSharedEndpointsall'API di Amazon Simple Storage Service.

Example : creazione di un endpoint

Nell'esempio seguente viene creato un endpoint per un Outpost condiviso. Prima di utilizzare questo comando, sostituisci i valori *user input placeholder* per l'ID dell'Outpost, l'ID della sottorete e l'ID del gruppo di sicurezza con i valori appropriati per il tuo caso d'uso.

> **Note**
>
> L'utente può eseguire questa operazione solo se la condivisione di risorse include la risorsa Outposts.

```
aws s3outposts create-endpoint --outposts-id op-01ac5d28a6a232904 --subnet-id XXXXXX --
security-group-id XXXXXXX
```

Per ulteriori informazioni su questo comando, consultare create-endpoint nella Guida di riferimento di AWS CLI .

Altri Servizi AWS che utilizzano S3 su Outposts

Gli altri Servizi AWS che vengono eseguiti localmente in AWS Outposts possono anche utilizzare la capacità di Amazon S3 su Outposts. In Amazon CloudWatch lo spazio dei nomi `S3Outposts` mostra i parametri dettagliati dei bucket in S3 su Outposts che tuttavia non includono l'utilizzo per altri Servizi AWS. Per gestire la capacità di S3 su Outposts che viene utilizzata da altri Servizi AWS, consulta le informazioni nella tabella seguente.

Servizio AWS	Descrizione	Ulteriori informazioni
Amazon S3	Tutto l'utilizzo diretto di S3 on Outposts ha un parametro CloudWatch di bucket e account corrispondente.	Vedi i parametri
Amazon Elastic Block Store (Amazon EBS)	Per Amazon EBS su Outposts, puoi scegliere un AWS Outpost come destinazione snapshot e archiviare localmente nel tuo S3 su Outpost.	Ulteriori informazioni
Amazon Relational Database Service (Amazon RDS)	Puoi utilizzare i backup locali di Amazon RDS per archiviare i backup RDS localmente sul tuo Outpost.	Ulteriori informazioni

Monitoraggio di S3 su Outposts

Con Amazon S3 on Outposts, puoi creare bucket S3 sui tuoi AWS Outposts e archiviare e recuperare facilmente oggetti in locale per applicazioni che richiedono accesso locale ai dati, elaborazione locale dei dati e residenza dei dati. S3 on Outposts offre una nuova classe di storage, S3 Outposts `OUTPOSTS` (), che utilizza le API di Amazon S3 ed è progettata per archiviare i dati in modo duraturo e ridondante su più dispositivi e server sul tuo. AWS Outposts Comunichi con il bucket Outpost utilizzando un punto di accesso e una connessione di endpoint su un Virtual Private Cloud (VPC). Sui bucket Outposts puoi utilizzare le stesse API e caratteristiche di Amazon S3, comprese policy di accesso, crittografia e tagging. Puoi usare S3 su Outposts tramite AWS Management Console AWS Command Line Interface ,AWS CLI() AWS , SDK o API REST. Per ulteriori informazioni, consulta Che cos'è Amazon S3 su Outposts?

Per ulteriori informazioni su come monitorare la capacità di archiviazione di Amazon S3 su Outposts, consulta i seguenti argomenti.

Argomenti

Gestione della capacità di S3 on Outposts con i parametri di Amazon CloudWatch

Per aiutarti a gestire la capacità fissa di S3 su Outpost, ti consigliamo di creare CloudWatch avvisi che ti avvisino quando l'utilizzo dello storage supera una determinata soglia. Per ulteriori informazioni sulle CloudWatch metriche per S3 su Outposts, consulta. CloudWatch metriche Se non c'è spazio sufficiente per archiviare un oggetto nell'outpost, l'API restituirà una ICE, ovvero una esenzione da capacità insufficiente. Per liberare spazio, puoi creare CloudWatch allarmi che attivano l'eliminazione esplicita dei dati o utilizzare una politica di scadenza del ciclo di vita per far scadere gli oggetti. Per salvare i dati prima dell'eliminazione, puoi copiare AWS DataSync i dati dal tuo bucket Amazon S3 on Outposts a un bucket S3 in un. Regione AWSPer ulteriori informazioni sull'utilizzo DataSync, consulta Getting Started with AWS DataSync nella Guida per l'utente.AWS DataSync

CloudWatch metriche

Lo spazio dei nomi `S3Outposts` include i seguenti parametri per i bucket Amazon S3 on Outposts. È possibile monitorare il numero totale di byte S3 on Outposts di cui è stato eseguito il provisioning, il totale dei byte liberi disponibili per gli oggetti e la dimensione totale di tutti gli oggetti per un determinato bucket. Esistono parametri relativi a bucket o account per tutti gli usi diretti di S3. L'utilizzo indiretto di S3, ad esempio l'archiviazione di snapshot locali di Amazon Elastic Block Store o dei backup di Amazon Relational Database Service su Outposts, consuma la capacità di S3, ma non è incluso nei parametri relativ a bucket o account. Per ulteriori informazioni sugli snapshot locali di Amazon EBS, consulta Snapshot locali di Amazon EBS su Outposts. Per visualizzare il report sui costi di Amazon EBS, visita https://console.aws.amazon.com/billing/.

> **Note**
>
> S3 su Outposts supporta solo i parametri riportati di seguito e non supporta altri parametri Amazon S3.

Poiché S3 on Outposts ha un limite di capacità fisso, ti consigliamo di CloudWatch creare allarmi per avvisarti quando l'utilizzo dello storage supera una determinata soglia.

Parametro	Descrizione	Periodo di tempo	Unità	Type
`Outpost talByte`	La capacità totale di cui è stato eseguito il provisioning in byte per un outpost.	5 minuti	Byte	S3 on Outposts
`Outpost eeBytes`	Il numero di byte gratuiti disponibili in un Outpost per archiviare i dati dei clienti.	5 minuti	Byte	S3 on Outposts
`BucketU dBytes`	La dimensione totale di tutti gli oggetti per il bucket specifica to.	5 minuti	Byte	S3 su Outposts. Solo utilizzo diretto di S3.
`Accountl edBytes`	La dimensione totale di tutti gli oggetti per l'account Outposts specificato.	5 minuti	Byte	S3 su Outposts. Solo utilizzo diretto di S3.
`BytesPe ingRepl ation`	Numero totale di byte di oggetti in attesa di replica per una determinata regola di replica. Per ulteriori informazi oni su come abilitare le metriche di replica, consulta l'argomento relativo alla creazione di regole di replica tra outpost.	5 minuti	Byte	Facoltativo. Per Replica Amazon S3 su Outposts.
`Operati sPendin`	Numero totale di operazion i in attesa di replica per una determinata regola di replica. Per ulteriori informazioni su	5 minuti	Conteggi	Facoltativo. Per Replica Amazon S3 su Outposts.

Parametro	Descrizione	Periodo di tempo	Unità	Type
`eplicat:` n	come abilitare le metriche di replica, consulta l'argomento relativo alla creazione di regole di replica tra outpost.			
`Replica:` `onLaten(`	Numero corrente di secondi entro i quali i bucket di destinazione della replica sono in ritardo rispetto al bucket di origine per una determinata regola di replica. Per ulteriori informazioni su come abilitare le metriche di replica, consulta l'argomento relativo alla creazione di regole di replica tra outpost.	5 minuti	Secondi	Facoltativo. Per Replica Amazon S3 su Outposts.

Ricezione di notifiche sugli eventi di S3 on Outposts utilizzando Amazon Events CloudWatch

Puoi utilizzare CloudWatch Events per creare una regola per qualsiasi evento API Amazon S3 on Outposts. Quando crei una regola, puoi scegliere di ricevere una notifica tramite tutte le CloudWatch destinazioni supportate, tra cui Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS) e. AWS Lambda Per ulteriori informazioni, consulta l'elenco dei AWS servizi che possono essere utilizzati come target per CloudWatch Events nella Amazon CloudWatch Events User Guide. Per scegliere un servizio di destinazione da utilizzare con il tuo S3 su Outposts, consulta Creazione di CloudWatch una regola Events che si attiva su AWS una chiamata API utilizzata AWS CloudTrail nella CloudWatch Amazon Events User Guide.

> **Note**
>
> Per le operazioni sugli oggetti S3 on Outposts AWS , gli eventi di chiamata API inviati CloudTrail da corrisponderanno alle tue regole solo se hai trail (opzionalmente con selettori

di eventi) configurati per ricevere tali eventi. Per ulteriori informazioni, consulta Lavorare con i file di CloudTrail registro nella Guida per l'utente.AWS CloudTrail

Example

Di seguito è riportata una regola di esempio per l'operazione `DeleteObject`. Per utilizzare questa regola di esempio, sostituisci *`example-s3-bucket1`* con il nome del bucket S3 su Outposts.

```
{
  "source": [
    "aws.s3-outposts"
  ],
  "detail-type": [
    "AWS API call through CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "s3-outposts.amazonaws.com"
    ],
    "eventName": [
      "DeleteObject"
    ],
    "requestParameters": {
      "bucketName": [
        "example-s3-bucket1"
      ]
    }
  }
}
```

Monitoraggio di S3 su Outposts con i log AWS CloudTrail

Amazon S3 on Outposts è integrato con AWS CloudTrail un servizio che fornisce un registro delle azioni intraprese da un utente, un ruolo o un utente in Servizio AWS S3 on Outposts. Puoi utilizzare AWS CloudTrail per ottenere informazioni relative alle richieste S3 su Outposts a livello di bucket e a livello di oggetto per controllare e registrare l'attività degli eventi di S3 su Outposts. Per abilitare gli eventi CloudTrail relativi ai dati per tutti i bucket Outposts o per un elenco di bucket Outposts specifici, devi creare un percorso manualmente in. CloudTrail Per ulteriori informazioni sulle voci dei file di CloudTrail registro, consulta S3 sulle voci dei file di registro di Outposts.

Note

- È consigliabile creare una politica del ciclo di vita per il bucket Outposts degli eventi AWS CloudTrail relativi ai dati. Configura la policy del ciclo di vita in modo tale da rimuovere periodicamente i file di log al termine del periodo di tempo desiderato per l'audit. In questo modo, si riduce la quantità di dati analizzati da Amazon Athena per ogni query. Per ulteriori informazioni, consulta Impostazione di una configurazione del ciclo di vita su un bucket.
- Per esempi su come interrogare CloudTrail i log, consulta il post sul blog AWS Big Data Analyze Security, Compliance, and Operational Activity Using AWS CloudTrail and Amazon Athena.

Abilita CloudTrail la registrazione degli oggetti in un bucket S3 on Outposts

Puoi utilizzare la console Amazon S3 per configurare un AWS CloudTrail trail per registrare gli eventi relativi ai dati per gli oggetti in un bucket Amazon S3 on Outposts. CloudTrail supporta la registrazione di S3 su operazioni API a livello di oggetto Outposts come, e. `GetObject` `DeleteObject` `PutObject` Questi eventi vengono chiamati eventi di dati.

Per impostazione predefinita, i CloudTrail trail non registrano gli eventi relativi ai dati. Puoi tuttavia configurare i trail per registrare gli eventi di dati per i bucket S3 su Outposts specificati oppure per registrare gli eventi di dati per tutti i bucket S3 su Outposts nel tuo Account AWS. Per ulteriori informazioni, consulta Registrazione delle chiamate API Amazon S3 tramite AWS CloudTrail.

CloudTrail non inserisce gli eventi relativi ai dati nella cronologia degli CloudTrail eventi. Inoltre, non tutte le operazioni API a livello di bucket di S3 on Outposts sono inserite nella cronologia degli eventi. CloudTrail Per ulteriori informazioni su come interrogare CloudTrail i log, consulta Usare i modelli di filtro di Amazon CloudWatch Logs e Amazon Athena per CloudTrail interrogare i log nel Knowledge Center. AWS

Per configurare un trail per registrare gli eventi di dati per un bucket S3 su Outposts, puoi utilizzare la console AWS CloudTrail o la console Amazon S3. Se stai configurando un percorso per registrare gli eventi relativi ai dati per tutti i bucket S3 on Outposts del tuo Account AWS, è più facile usare la console. CloudTrail Per informazioni sull'utilizzo della CloudTrail console per configurare un percorso per registrare gli eventi relativi ai dati di S3 in Outposts, consulta Data events AWS CloudTrail nella User Guide.

> **Important**
>
> Per gli eventi di dati sono previsti costi aggiuntivi. Per ulteriori informazioni, consulta Prezzi di AWS CloudTrail.

La procedura seguente mostra come utilizzare la console Amazon S3 per configurare un CloudTrail trail per registrare gli eventi relativi ai dati per un bucket S3 on Outposts.

> **Note**
>
> Chi Account AWS crea il bucket lo possiede ed è l'unico che può configurare gli eventi dati di S3 on Outposts a cui inviare. AWS CloudTrail

Per abilitare la registrazione degli eventi CloudTrail relativi ai dati per gli oggetti in un bucket S3 on Outposts

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel riquadro di navigazione a sinistra, seleziona Outposts buckets (Bucket Outposts).
3. Scegli il nome del bucket Outposts di cui desideri utilizzare gli eventi relativi ai dati. CloudTrail
4. Scegli Properties (Proprietà).
5. Nella sezione relativa agli eventi AWS CloudTrail relativi ai dati, scegli Configura in. CloudTrail

   La AWS CloudTrail console si apre.

   Puoi creare un nuovo CloudTrail percorso o riutilizzare un percorso esistente e configurare gli eventi dati di S3 on Outposts in modo che vengano registrati nel tuo percorso.
6. Nella pagina Dashboard della CloudTrail console, scegli Crea percorso.
7. Nella pagina Fase 1 - Seleziona attributi trail, specifica un nome per il percorso, scegli un bucket S3 per archiviare i log del trail, specifica le altre impostazioni desiderate e quindi scegli Avanti.
8. Nella pagina Fase 2 - Seleziona eventi di log, in Tipo di evento, scegli Eventi di dati.

   In Tipo di evento di dati, scegli S3 Outposts. Seleziona Successivo.

> **Note**
>
> - Quando crei un trail e configuri la registrazione degli eventi di dati per S3 su Outposts, devi specificare correttamente il tipo di evento di dati.
>   - Se usi la CloudTrail console, scegli il tipo di evento S3 Outposts for Data. Per informazioni su come creare percorsi nella CloudTrail console, consulta Creazione e aggiornamento di un percorso con la console nella Guida per l'AWS CloudTrail utente. Per informazioni su come configurare S3 sulla registrazione degli eventi dati di Outposts nella CloudTrail console, consulta Logging data events for Amazon S3 Objects nella User Guide.AWS CloudTrail
>   - Se usi AWS Command Line Interface (AWS CLI) o gli AWS SDK, imposta il campo su. `resources.type` `AWS::S3Outposts::Object` Per ulteriori informazioni su come registrare gli eventi dati di S3 su Outposts con AWS CLI, consulta Log S3 on Outposts nella Guida per l'utente.AWS CloudTrail
> - Se utilizzi la CloudTrail console o la console Amazon S3 per configurare un percorso per registrare gli eventi relativi ai dati per un bucket S3 on Outposts, la console Amazon S3 mostra che la registrazione a livello di oggetto è abilitata per il bucket.

9. Nella pagina Fase 3 - Verifica e crea, rivedi gli attributi del trail e i log eventi che hai configurato. Quindi scegli Crea trail.

Per disabilitare la registrazione degli eventi CloudTrail relativi ai dati per gli oggetti in un bucket S3 on Outposts

1. Accedi AWS Management Console e apri la console all'indirizzo https://console.aws.amazon.com/cloudtrail/. CloudTrail
2. Nel pannello di navigazione a sinistra, scegli Trail.
3. Scegli il nome del trail che hai creato per registrare i log eventi del bucket S3 su Outposts.
4. Nella pagina dei dettagli del trail, scegli Interrompi la registrazione nell'angolo in alto a destra.
5. Nella finestra di dialogo visualizzata, scegli Interrompi la registrazione.

Sviluppo con Amazon S3 su Outposts

Con Amazon S3 su Outposts è possibile creare bucket S3 su AWS Outposts, nonché archiviare e recuperare facilmente gli oggetti on-Premise per le applicazioni che richiedono l'accesso ai dati in locale, l'elaborazione dei dati in locale e la residenza dei dati. S3 su Outposts fornisce una nuova classe di archiviazione, S3 Outposts (OUTPOSTS), che utilizza le API Amazon S3 ed è progettata per archiviare i dati in modo durevole e ridondante su più dispositivi e server su AWS Outposts. Comunichi con il bucket Outpost utilizzando un punto di accesso e una connessione di endpoint su un Virtual Private Cloud (VPC). Sui bucket Outposts puoi utilizzare le stesse API e caratteristiche di Amazon S3, comprese policy di accesso, crittografia e tagging. Puoi utilizzare S3 su Outposts tramite la AWS Management Console, AWS Command Line Interface (AWS CLI), gli SDK AWS o l'API REST. Per ulteriori informazioni, consulta Che cos'è Amazon S3 su Outposts?

I seguenti argomenti forniscono informazioni sullo sviluppo con S3 su Outposts

Argomenti

- Operazioni API in Amazon S3 su Outposts
- Configurazione del client di controllo S3 per S3 su Outposts utilizzando SDK per Java
- Effettuare richieste a S3 su Outposts tramite IPv6

Operazioni API in Amazon S3 su Outposts

In questo argomento viene fornito l'elenco delle operazioni API su Amazon S3, Amazon S3 Control e Amazon S3 su Outposts che puoi utilizzare in Amazon S3 su Outposts.

Argomenti

- Operazioni API Amazon S3 per la gestione degli oggetti
- Operazioni API Amazon S3 Control per la gestione dei bucket
- Operazioni API S3 su Outposts per la gestione di Outposts

Operazioni API Amazon S3 per la gestione degli oggetti

S3 su Outposts è progettato per utilizzare le stesse operazioni API sugli oggetti di Amazon S3. È necessario utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outpost. Quando utilizzi un'operazione API di oggetti con S3 su Outposts, fornisci il nome della risorsa Amazon (ARN) del punto di accesso Outposts o l'alias del punto di accesso. Per ulteriori informazioni

sugli alias del punto di accesso, consulta Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3 su Outposts.

Amazon S3 su Outposts supporta le seguenti operazioni API Amazon S3:

- AbortMultipartUpload
- CompleteMultipartUpload
- CopyObject
- CreateMultipartUpload
- DeleteObject
- DeleteObjects
- DeleteObjectTagging
- GetObject
- GetObjectTagging
- HeadBucket
- HeadObject
- ListMultipartUploads
- ListObjects
- ListObjectsV2
- ListObjectVersions
- ListParts
- PutObject
- PutObjectTagging
- UploadPart
- UploadPartCopy

Operazioni API Amazon S3 Control per la gestione dei bucket

S3 su Outposts supporta le seguenti operazioni API Amazon S3 Control per le operazioni sui bucket.

- 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

Operazioni API S3 su Outposts per la gestione di Outposts

S3 su Outposts supporta le seguenti operazioni API Amazon S3 su Outposts per la gestione degli endpoint.

- CreateEndpoint
- DeleteEndpoint
- ListEndpoints

- ListOutpostsWithS3
- ListSharedEndpoints

Configurazione del client di controllo S3 per S3 su Outposts utilizzando SDK per Java

Nell'esempio seguente viene configurato il client di controllo Amazon S3 per Amazon S3 su Outposts mediante AWS SDK for Java. Per utilizzare questo comando, sostituisci *`user input placeholder`* con le tue informazioni.

```
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();

}
```

Effettuare richieste a S3 su Outposts tramite IPv6

Gli endpoint dual-stack Amazon S3 on Outposts e S3 on Outposts supportano le richieste ai bucket S3 on Outposts utilizzando il protocollo IPv6 o IPv4. Con il supporto IPv6 per S3 on Outposts, puoi accedere e gestire i tuoi bucket e controllare le risorse del piano tramite le API S3 on Outposts su reti IPv6.

> **Note**
>
> Le azioni degli oggetti S3 on Outposts (`PutObject`come `GetObject` o) non sono supportate sulle reti IPv6.

Non sono previsti costi aggiuntivi per l'accesso a S3 on Outposts su reti IPv6. Per ulteriori informazioni su S3 on Outposts, consulta i prezzi di S3 on Outposts.

Argomenti

Nozioni di base su IPv6

Per effettuare una richiesta a un bucket S3 on Outposts su IPv6, devi utilizzare un endpoint dual-stack. Nella sezione seguente viene descritto come effettuare richieste su IPv6 utilizzando gli endpoint dual-stack.

Le seguenti sono considerazioni importanti prima di provare ad accedere a un bucket S3 on Outposts tramite IPv6:

- Il client e la rete che eseguono l'accesso al bucket devono essere abilitati a utilizzare IPv6.
- Per l'accesso a IPv6 sono supportate sia le richieste in stile percorso sia quelle in stile hosting virtuale. Per ulteriori informazioni, consulta Utilizzo di S3 sugli endpoint dual-stack Outposts.
- Se utilizzi il filtraggio degli indirizzi IP di origine nel tuo utente AWS Identity and Access Management (IAM) o le policy del bucket S3 on Outposts, devi aggiornare le politiche per includere gli intervalli di indirizzi IPv6.

  > **Note**
  >
  > Questo requisito si applica solo alle operazioni dei bucket S3 on Outposts e alle risorse del piano di controllo su reti IPv6. Le azioni oggetto di Amazon S3 on Outposts non sono supportate sulle reti IPv6.

- Quando si utilizza IPv6, i file di log degli accessi al server generano indirizzi IP in un formato IPv6. È necessario aggiornare gli strumenti, gli script e il software esistenti utilizzati per analizzare i file di registro di S3 su Outposts, in modo che possano analizzare gli indirizzi IP remoti in formato IPv6.

Gli strumenti, gli script e il software aggiornati analizzeranno quindi correttamente gli indirizzi IP remoti in formato IPv6.

Utilizzo di endpoint dual-stack per effettuare richieste su una rete IPv6

Per effettuare richieste con S3 on Outposts API su IPv6, puoi utilizzare endpoint dual-stack tramite il nostro SDK. AWS CLI AWS Le operazioni dell'API di controllo Amazon S3 e le operazioni dell'API S3 on Outposts funzionano allo stesso modo indipendentemente dal fatto che accedi a S3 on Outposts tramite un protocollo IPv6 o un protocollo IPv4. Tuttavia, tieni presente che le azioni degli oggetti S3 on Outposts (`PutObject`come `GetObject` o) non sono supportate sulle reti IPv6.

Quando si utilizzano AWS Command Line Interface (AWS CLI) e gli SDK AWS, è possibile servirsi di un parametro o flag per passare a un endpoint dual-stack. Puoi anche specificare l'endpoint dual-stack direttamente come override dell'endpoint S3 on Outposts nel file di configurazione.

Puoi utilizzare un endpoint dual-stack per accedere a un bucket S3 on Outposts tramite IPv6 da uno dei seguenti:

- La AWS CLI, consulta Utilizzo degli endpoint dual-stack dall AWS CLI.
- Gli SDK AWS, consulta Utilizzo di S3 sugli endpoint dual-stack Outposts dagli SDK AWS.

Utilizzo degli indirizzi IPv6 nelle policy IAM

Prima di provare ad accedere a un bucket S3 on Outposts utilizzando un protocollo IPv6, assicurati che gli utenti IAM o le policy dei bucket S3 on Outposts utilizzate per il filtraggio degli indirizzi IP siano aggiornate per includere gli intervalli di indirizzi IPv6. Se i criteri di filtraggio degli indirizzi IP non vengono aggiornati per gestire gli indirizzi IPv6, puoi perdere l'accesso a un bucket S3 on Outposts mentre provi a utilizzare il protocollo IPv6.

Le politiche IAM che filtrano gli indirizzi IP utilizzano operatori di condizione degli indirizzi IP. La seguente policy sui bucket di S3 on Outposts identifica l'intervallo IP 54.240.143.* di indirizzi IPv4 consentiti utilizzando gli operatori di condizione dell'indirizzo IP. A qualsiasi indirizzo IP al di fuori di questo intervallo verrà negato l'accesso al `DOC-EXAMPLE-BUCKET` bucket S3 on Outposts (). Poiché tutti gli indirizzi IPv6 non sono inclusi nell'intervallo consentito, questa policy impedisce agli indirizzi IPv6 di accedere 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"}
      }
    }
  ]
}
```

Puoi modificare l'elemento della policy del bucket S3 on Outposts per consentire sia gli `Condition` intervalli di indirizzi IPv4 `54.240.143.0/24` () che IPv6 `2001:DB8:1234:5678::/64` (), come mostrato nell'esempio seguente. È possibile utilizzare lo stesso tipo di blocco `Condition` mostrato nell'esempio per aggiornare sia le policy del bucket sia le policy utente IAM.

```
       "Condition": {
         "IpAddress": {
            "aws:SourceIp": [
              "54.240.143.0/24",
               "2001:DB8:1234:5678::/64"
             ]
          }
        }
```

Prima di utilizzare IPv6, è necessario aggiornare tutte le policy del bucket e utente IAM pertinenti che utilizzano il filtro degli indirizzi IP per consentire gli intervalli di indirizzi IPv6. Ti consigliamo di aggiornare le policy IAM con gli intervalli di indirizzi IPv6 dell'organizzazione oltre agli intervalli di indirizzi IPv4 esistenti. Per un esempio di una policy del bucket che consenta l'accesso sia su IPv6 sia su IPv4, consulta Limitare l'accesso a indirizzi IP specifici.

È possibile esaminare le policy utente IAM tramite la console IAM all'indirizzo https://console.aws.amazon.com/iam/. Per ulteriori informazioni su IAM, consulta la Guida per l'utente di IAM. Per informazioni sulla modifica delle politiche relative ai bucket di S3 on Outposts, consulta. Aggiunta o modifica di una policy di un bucket Amazon S3 su Outposts

Test di compatibilità degli indirizzi IP

Se utilizzi un'istanza Linux o Unix o una piattaforma macOS X, puoi testare il tuo accesso a un endpoint dual-stack tramite IPv6. Ad esempio, per testare la connessione ad Amazon S3 sugli endpoint Outposts su IPv6, usa il comando: `dig`

```
dig s3-outposts.us-west-2.api.aws AAAA +short
```

Se l'endpoint dual-stack su una rete IPv6 è configurato correttamente, il comando restituisce gli indirizzi IPv6 connessi. `dig` Per esempio:

```
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
```

Utilizzo di IPv6 con AWS PrivateLink

S3 on Outposts supporta il protocollo AWS PrivateLink IPv6 per servizi ed endpoint. Con il AWS PrivateLink supporto per il protocollo IPv6, puoi connetterti agli endpoint di servizio all'interno del tuo VPC tramite reti IPv6, da connessioni locali o da altre connessioni private. Il supporto IPv6 AWS PrivateLinkper S3 on Outposts consente anche l'integrazione con endpoint dual-stack. AWS PrivateLink Per istruzioni su come abilitare IPv6 per, consulta Accelerare l'adozione dell'IPv6 con servizi ed AWS PrivateLink endpoint. AWS PrivateLink

> **Note**
>
> Per aggiornare il tipo di indirizzo IP supportato da IPv4 a IPv6, consulta Modificare il tipo di indirizzo IP supportato nella Guida per l'utente. AWS PrivateLink

Utilizzo di IPv6 con AWS PrivateLink

Se utilizzi AWS PrivateLink con IPv6, devi creare un endpoint di interfaccia IPv6 o VPC dual-stack. Per i passaggi generali su come creare un endpoint VPC utilizzando ilAWS Management Console, consulta Accedere a un AWS servizio utilizzando un endpoint VPC di interfaccia nella Guida per l'utente. AWS PrivateLink

AWS Management Console

Usa la seguente procedura per creare un endpoint VPC di interfaccia che si connette a S3 su Outposts.

1. Accedi alla AWS Management Console e apri la console VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Nel pannello di navigazione, seleziona Endpoints (Endpoint).
3. Seleziona Crea endpoint.
4. Per Service category (Categoria servizio), scegli AWS services.
5. Per il nome del servizio, scegli il servizio S3 on Outposts (com.amazonaws.us-east-1.s3-outposts).
6. Per VPC, scegli il VPC da cui accederai a S3 su Outposts.
7. Per le sottoreti, scegli una sottorete per zona di disponibilità da cui accedere a S3 su Outposts. Non è possibile selezionare più sottoreti dalla stessa zona di disponibilità. Per ogni sottorete selezionata, viene creata una nuova interfaccia di rete endpoint. Per impostazione predefinita, gli indirizzi IP degli intervalli di indirizzi IP della sottorete vengono assegnati alle interfacce di rete degli endpoint. Per designare un indirizzo IP per un'interfaccia di rete endpoint, scegli Designare indirizzi IP e inserisci un indirizzo IPv6 dall'intervallo di indirizzi di sottorete.
8. Per il tipo di indirizzo IP, scegli Dualstack. Assegna indirizzi IPv4 e IPv6 alle interfacce di rete degli endpoint. Questa opzione è supportata solo se tutte le sottoreti selezionate dispongono di intervalli di indirizzi IPv4 e IPv6.
9. Per i gruppi di sicurezza, scegli i gruppi di sicurezza da associare alle interfacce di rete degli endpoint per l'endpoint VPC. Per impostazione predefinita, il gruppo di sicurezza predefinito è associato al VPC.
10. Per Policy, scegli Accesso completo per consentire a tutti i principali di eseguire tutte le operazioni su tutte le risorse dell'endpoint VPC. Altrimenti, scegli Personalizzato per allegare una policy degli endpoint VPC che controlli le autorizzazioni di cui dispongono i responsabili per eseguire azioni sulle risorse sull'endpoint VPC. Questa opzione è disponibile solo se il servizio supporta le policy dell'endpoint VPC. Per ulteriori informazioni, consulta le politiche degli endpoint.
11. (Facoltativo) Per aggiungere un tag, scegliere Add new tag (Aggiungi nuovo tag) e immettere la chiave e il valore del tag.
12. Seleziona Crea endpoint.

Example — Politica sui bucket di S3 on Outposts

Per consentire a S3 on Outposts di interagire con i tuoi endpoint VPC, puoi aggiornare la tua policy S3 on Outposts in questo modo:

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3-outposts:*",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

AWS CLI

> **Note**
>
> Per abilitare la rete IPv6 sul tuo endpoint VPC, devi aver `IPv6` impostato il filtro per S3 su `OutpostsSupportedIpAddressType`.

L'esempio seguente utilizza il `create-vpc-endpoint` comando per creare un nuovo endpoint di interfaccia dual-stack.

```
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"
```

A seconda della configurazione del AWS PrivateLink servizio, potrebbe essere necessario accettare le connessioni endpoint appena create dal provider di servizi endpoint VPC prima di poter essere utilizzate. Per ulteriori informazioni, consulta Accettare e rifiutare le richieste di connessione agli endpoint nella Guida per l'utente. AWS PrivateLink

L'esempio seguente utilizza il `modify-vpc-endpoint` comando per aggiornare l'endpoint VPC solo IPV a un endpoint dual-stack. L'endpoint dual-stack consente l'accesso a entrambe le reti 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"
```

Per ulteriori informazioni su come abilitare la rete IPv6 per, consulta Accelerare l'adozione dell'IPv6 con servizi ed AWS PrivateLink endpoint. AWS PrivateLink

Utilizzo di S3 sugli endpoint dual-stack Outposts

Gli endpoint dual-stack S3 on Outposts supportano le richieste ai bucket S3 on Outposts su IPv6 e IPv4. Questa sezione descrive come usare S3 sugli endpoint dual-stack Outposts.

Argomenti

- Endpoint dual-stack S3 on Outposts
- Utilizzo degli endpoint dual-stack dall AWS CLI
- Utilizzo di S3 sugli endpoint dual-stack Outposts dagli SDK AWS

Endpoint dual-stack S3 on Outposts

Quando effettui una richiesta a un endpoint dual-stack, l'URL del bucket S3 on Outposts si risolve in un indirizzo IPv6 o IPv4. Per ulteriori informazioni sull'accesso a un bucket S3 on Outposts tramite IPv6, consulta. Effettuare richieste a S3 su Outposts tramite IPv6

Per accedere a un bucket S3 on Outposts tramite un endpoint dual-stack, usa un nome di endpoint in stile path. S3 on Outposts supporta solo nomi di endpoint dual-stack regionali, il che significa che devi specificare la regione come parte del nome.

Per un endpoint FIPS in stile percorso dual-stack, usa la seguente convenzione di denominazione:

```
s3-outposts-fips.region.api.aws
```

Per un endpoint dual-stack non FIPS, utilizzate la seguente convenzione di denominazione:

```
s3-outposts.region.api.aws
```

> **Note**
>
> I nomi degli endpoint in stile host virtuale non sono supportati in S3 on Outposts.

Utilizzo degli endpoint dual-stack dall AWS CLI

In questa sezione vengono forniti esempi dei comandi dell'AWS CLI utilizzati per effettuare le richieste a un endpoint dual-stack. Per istruzioni sull'impostazione di AWS CLI, consulta Guida introduttiva all'utilizzo di AWS CLI and SDK for Java.

Imposta il valore `use_dualstack_endpoint` di configurazione su un profilo nel tuo AWS Config file per indirizzare tutte le richieste Amazon S3 effettuate dai `s3api` AWS CLI comandi `s3` and all'endpoint dual-stack per la regione specificata. `true` Specificate la regione nel file di configurazione o in un comando utilizzando l'opzione. `--region`

Quando si utilizzano endpoint dual-stack conAWS CLI, è supportato solo lo stile di `path` indirizzamento. Lo stile di indirizzamento, impostato nel file di configurazione, determina se il nome del bucket si trova nel nome host o nell'URL. Per ulteriori informazioni, consulta s3outposts nella Guida per l'utente di AWS CLI.

Per utilizzare un endpoint dual-stack tramite ilAWS CLI, usa il `--endpoint-url` parametro con l'endpoint `http://s3.dualstack.region.amazonaws.com` o `https://s3-outposts-fips.region.api.aws` per qualsiasi comando. `s3control` `s3outposts`

Per esempio:

```
$  aws s3control list-regional-buckets --endpoint-url https://s3-
outposts.region.api.aws
```

Utilizzo di S3 sugli endpoint dual-stack Outposts dagli SDK AWS

In questa sezione vengono forniti esempi su come accedere all'endpoint dual-stack tramite gli SDK AWS.

AWS SDK for Java 2.xEsempio di endpoint Dual-Stack con

Gli esempi seguenti mostrano come utilizzare le `S3OutpostsClient` classi `S3ControlClient` and per abilitare gli endpoint dual-stack durante la creazione di un client S3 su Outposts utilizzando. AWS SDK for Java 2.x Per istruzioni su come creare e testare un esempio Java funzionante per Amazon S3 su Outposts, consulta. Guida introduttiva all'utilizzo di AWS CLI and SDK for Java

Example — Crea una **`S3ControlClient`** classe con endpoint dual-stack abilitati

```
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 — Crea una con endpoint dual-stack **`S3OutpostsClient`** abilitati

```
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();
        }
    }
}
```

Se utilizzi Windows, potresti dover impostare la AWS SDK for Java 2.x seguente proprietà della macchina virtuale Java (JVM):

```
java.net.preferIPv6Addresses=true
```

Esempi di codice per Amazon S3 che utilizzano SDK AWS

I seguenti esempi di codice mostrano come usare Amazon S3 con un kit di sviluppo AWS software (SDK).

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Esempi cross-service: applicazioni di esempio che funzionano su più servizi Servizi AWS.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Nozioni di base

Hello Amazon S3

Gli esempi di codice seguenti mostrano come iniziare a utilizzare Amazon S3.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Codice per il file CMake C MakeLists .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})
```

Codice per il file origine 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;
}
```

- Per i dettagli sull'API, consulta API ListBucketsReference AWS SDK for C++ .

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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)
  }
 }
}
```

- Per i dettagli sull'API, ListBucketsconsulta AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
}
```

- Per i dettagli sull'API, ListBucketsconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
};
```

- Per i dettagli sull'API, ListBucketsconsulta AWS SDK for JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
use Aws\S3\S3Client;
```

```
$client = new S3Client(['region' => 'us-west-2']);
$results = $client->listBuckets();
var_dump($results);
```

- Per i dettagli sull'API, ListBucketsconsulta AWS SDK for PHP API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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()
```

- Per i dettagli sull'API, consulta ListBuckets AWSSDK for Python (Boto3) API Reference.

Esempi di codice

- Utilizzo GetBucketLocation con un AWS SDK o una CLI
- Utilizzo GetBucketLogging con un AWS SDK o una CLI
- Utilizzo GetBucketMetricsConfiguration con un AWS SDK o una CLI
- Utilizzo GetBucketNotification con un AWS SDK o una CLI
- Utilizzo GetBucketPolicy con un AWS SDK o una CLI
- Utilizzo GetBucketPolicyStatus con un AWS SDK o una CLI
- Utilizzo GetBucketReplication con un AWS SDK o una CLI
- Utilizzo GetBucketRequestPayment con un AWS SDK o una CLI
- Utilizzo GetBucketTagging con un AWS SDK o una CLI
- Utilizzo GetBucketVersioning con un AWS SDK o una CLI
- Utilizzo GetBucketWebsite con un AWS SDK o una CLI
- Utilizzo GetObject con un AWS SDK o una CLI
- Utilizzo GetObjectAcl con un AWS SDK o una CLI
- Utilizzo GetObjectLegalHold con un AWS SDK o una CLI
- Utilizzo GetObjectLockConfiguration con un AWS SDK o una CLI
- Utilizzo GetObjectRetention con un AWS SDK o una CLI
- Utilizzo GetObjectTagging con un AWS SDK o una CLI
- Utilizzo GetPublicAccessBlock con un AWS SDK o una CLI
- Utilizzo HeadBucket con un AWS SDK o una CLI
- Utilizzo HeadObject con un AWS SDK o una CLI
- Utilizzo ListBucketAnalyticsConfigurations con un AWS SDK o una CLI
- Utilizzo ListBucketInventoryConfigurations con un AWS SDK o una CLI
- Utilizzo ListBuckets con un AWS SDK o una CLI
- Utilizzo ListMultipartUploads con un AWS SDK o una CLI
- Utilizzo ListObjectVersions con un AWS SDK o una CLI
- Utilizzo ListObjects con un AWS SDK o una CLI
- Utilizzo ListObjectsV2 con un AWS SDK o una CLI
- Utilizzo PutBucketAccelerateConfiguration con un AWS SDK o una CLI
- Utilizzo PutBucketAcl con un AWS SDK o una CLI
- Utilizzo PutBucketCors con un AWS SDK o una CLI

- Utilizzo PutBucketEncryption con un AWS SDK o una CLI
- Utilizzo PutBucketLifecycleConfiguration con un AWS SDK o una CLI
- Utilizzo PutBucketLogging con un AWS SDK o una CLI
- Utilizzo PutBucketNotification con un AWS SDK o una CLI
- Utilizzo PutBucketNotificationConfiguration con un AWS SDK o una CLI
- Utilizzo PutBucketPolicy con un AWS SDK o una CLI
- Utilizzo PutBucketReplication con un AWS SDK o una CLI
- Utilizzo PutBucketRequestPayment con un AWS SDK o una CLI
- Utilizzo PutBucketTagging con un AWS SDK o una CLI
- Utilizzo PutBucketVersioning con un AWS SDK o una CLI
- Utilizzo PutBucketWebsite con un AWS SDK o una CLI
- Utilizzo PutObject con un AWS SDK o una CLI
- Utilizzo PutObjectAcl con un AWS SDK o una CLI
- Utilizzo PutObjectLegalHold con un AWS SDK o una CLI
- Utilizzo PutObjectLockConfiguration con un AWS SDK o una CLI
- Utilizzo PutObjectRetention con un AWS SDK o una CLI
- Utilizzo RestoreObject con un AWS SDK o una CLI
- Utilizzo SelectObjectContent con un AWS SDK o una CLI
- Utilizzo UploadPart con un AWS SDK o una CLI

- Scenari per Amazon S3 che utilizzano SDK AWS
  - Crea un URL predefinito per Amazon S3 utilizzando un SDK AWS
  - Una pagina Web che elenca gli oggetti Amazon S3 utilizzando un SDK AWS
  - Eliminare caricamenti multiparte incompleti su Amazon S3 utilizzando un SDK AWS
  - Scaricare tutti gli oggetti da un bucket Amazon Simple Storage Service (Amazon S3) in una directory locale
  - Ottieni un oggetto Amazon S3 da un punto di accesso multiregionale utilizzando un SDK AWS
  - Ottieni un oggetto da un bucket Amazon S3 utilizzando un AWS SDK, specificando un'intestazione If-Modified-Since
  - Inizia a usare bucket e oggetti Amazon S3 utilizzando un SDK AWS
  - Inizia a utilizzare la crittografia per oggetti Amazon S3 utilizzando un SDK AWS
  - Inizia a usare i tag per gli oggetti Amazon S3 utilizzando un SDK AWS

- Ottieni la configurazione di conservazione legale di un oggetto Amazon S3 utilizzando un SDK AWS
- Lavora con le funzionalità di blocco degli oggetti di Amazon S3 utilizzando un SDK AWS
- Gestisci gli elenchi di controllo degli accessi (ACL) per i bucket Amazon S3 utilizzando un SDK AWS
- Gestisci oggetti Amazon S3 con versioni in batch con una funzione Lambda utilizzando un SDK AWS
- Analizza gli URI di Amazon S3 utilizzando un SDK AWS
- Esegui una copia multiparte di un oggetto Amazon S3 utilizzando un SDK AWS
- Esegui un caricamento multiparte di un oggetto Amazon S3 utilizzando un SDK AWS
- Tieni traccia del caricamento o del download di un oggetto Amazon S3 utilizzando un SDK AWS
- Esempi di approcci per i test di unità e integrazione con un SDK AWS
- Caricare in modo ricorsivo una directory locale in un bucket Amazon Simple Storage Service (Amazon S3)
- Carica o scarica file di grandi dimensioni da e verso Amazon S3 utilizzando un SDK AWS
- Carica uno stream di dimensioni sconosciute su un oggetto Amazon S3 utilizzando un SDK AWS
- Usa i checksum per lavorare con un oggetto Amazon S3 utilizzando un SDK AWS
- Lavora con oggetti con versione di Amazon S3 utilizzando un SDK AWS

- Esempi serverless per Amazon S3 che utilizzano SDK AWS
  - Richiamo di una funzione Lambda da un trigger Amazon S3
- Esempi di servizi multipli per Amazon S3 che utilizzano SDK AWS
  - Creazione di un'app Amazon Transcribe
  - Convertire testo in voce e viceversa utilizzando un AWS SDK
  - Creazione di un'applicazione di gestione delle risorse fotografiche che consente agli utenti di gestire le foto utilizzando etichette
  - Creazione di un'applicazione Amazon Textract explorer
  - Rileva i DPI nelle immagini con Amazon Rekognition utilizzando un SDK AWS
  - Rileva le entità nel testo estratto da un'immagine utilizzando un SDK AWS
  - Rileva i volti in un'immagine utilizzando un SDK AWS
  - Rileva oggetti nelle immagini con Amazon Rekognition utilizzando un SDK AWS
  - Rileva persone e oggetti in un video con Amazon Rekognition utilizzando un SDK AWS

- Salva EXIF e altre informazioni sull'immagine utilizzando un SDK AWS
- Trasforma i dati per la tua applicazione con S3 Object Lambda

Azioni per Amazon S3 tramite SDK AWS

I seguenti esempi di codice mostrano come eseguire singole azioni Amazon S3 con AWS gli SDK. Questi estratti chiamano l'API Amazon S3 e sono estratti di codice di programmi più grandi che devono essere eseguiti in modo contestuale. Ogni esempio include un collegamento a GitHub, dove puoi trovare le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta la Documentazione di riferimento delle API di Amazon Simple Storage Service (Amazon S3).

Esempi

- Utilizzo AbortMultipartUpload con un AWS SDK o una CLI
- Utilizzo AbortMultipartUploads con un AWS SDK o una CLI
- Utilizzo CompleteMultipartUpload con un AWS SDK o una CLI
- Utilizzo CopyObject con un AWS SDK o una CLI
- Utilizzo CreateBucket con un AWS SDK o una CLI
- Utilizzo CreateMultiRegionAccessPoint con un AWS SDK o una CLI
- Utilizzo CreateMultipartUpload con un AWS SDK o una CLI
- Utilizzo DeleteBucket con un AWS SDK o una CLI
- Utilizzo DeleteBucketAnalyticsConfiguration con un AWS SDK o una CLI
- Utilizzo DeleteBucketCors con un AWS SDK o una CLI
- Utilizzo DeleteBucketEncryption con un AWS SDK o una CLI
- Utilizzo DeleteBucketInventoryConfiguration con un AWS SDK o una CLI
- Utilizzo DeleteBucketLifecycle con un AWS SDK o una CLI
- Utilizzo DeleteBucketMetricsConfiguration con un AWS SDK o una CLI
- Utilizzo DeleteBucketPolicy con un AWS SDK o una CLI
- Utilizzo DeleteBucketReplication con un AWS SDK o una CLI
- Utilizzo DeleteBucketTagging con un AWS SDK o una CLI
- Utilizzo DeleteBucketWebsite con un AWS SDK o una CLI

- Utilizzo DeleteObject con un AWS SDK o una CLI
- Utilizzo DeleteObjectTagging con un AWS SDK o una CLI
- Utilizzo DeleteObjects con un AWS SDK o una CLI
- Utilizzo DeletePublicAccessBlock con un AWS SDK o una CLI
- Utilizzo GetBucketAccelerateConfiguration con un AWS SDK o una CLI
- Utilizzo GetBucketAcl con un AWS SDK o una CLI
- Utilizzo GetBucketAnalyticsConfiguration con un AWS SDK o una CLI
- Utilizzo GetBucketCors con un AWS SDK o una CLI
- Utilizzo GetBucketEncryption con un AWS SDK o una CLI
- Utilizzo GetBucketInventoryConfiguration con un AWS SDK o una CLI
- Utilizzo GetBucketLifecycleConfiguration con un AWS SDK o una CLI
- Utilizzo GetBucketLocation con un AWS SDK o una CLI
- Utilizzo GetBucketLogging con un AWS SDK o una CLI
- Utilizzo GetBucketMetricsConfiguration con un AWS SDK o una CLI
- Utilizzo GetBucketNotification con un AWS SDK o una CLI
- Utilizzo GetBucketPolicy con un AWS SDK o una CLI
- Utilizzo GetBucketPolicyStatus con un AWS SDK o una CLI
- Utilizzo GetBucketReplication con un AWS SDK o una CLI
- Utilizzo GetBucketRequestPayment con un AWS SDK o una CLI
- Utilizzo GetBucketTagging con un AWS SDK o una CLI
- Utilizzo GetBucketVersioning con un AWS SDK o una CLI
- Utilizzo GetBucketWebsite con un AWS SDK o una CLI
- Utilizzo GetObject con un AWS SDK o una CLI
- Utilizzo GetObjectAcl con un AWS SDK o una CLI
- Utilizzo GetObjectLegalHold con un AWS SDK o una CLI
- Utilizzo GetObjectLockConfiguration con un AWS SDK o una CLI
- Utilizzo GetObjectRetention con un AWS SDK o una CLI
- Utilizzo GetObjectTagging con un AWS SDK o una CLI
- Utilizzo GetPublicAccessBlock con un AWS SDK o una CLI
- Utilizzo HeadBucket con un AWS SDK o una CLI

- Utilizzo HeadObject con un AWS SDK o una CLI
- Utilizzo ListBucketAnalyticsConfigurations con un AWS SDK o una CLI
- Utilizzo ListBucketInventoryConfigurations con un AWS SDK o una CLI
- Utilizzo ListBuckets con un AWS SDK o una CLI
- Utilizzo ListMultipartUploads con un AWS SDK o una CLI
- Utilizzo ListObjectVersions con un AWS SDK o una CLI
- Utilizzo ListObjects con un AWS SDK o una CLI
- Utilizzo ListObjectsV2 con un AWS SDK o una CLI
- Utilizzo PutBucketAccelerateConfiguration con un AWS SDK o una CLI
- Utilizzo PutBucketAcl con un AWS SDK o una CLI
- Utilizzo PutBucketCors con un AWS SDK o una CLI
- Utilizzo PutBucketEncryption con un AWS SDK o una CLI
- Utilizzo PutBucketLifecycleConfiguration con un AWS SDK o una CLI
- Utilizzo PutBucketLogging con un AWS SDK o una CLI
- Utilizzo PutBucketNotification con un AWS SDK o una CLI
- Utilizzo PutBucketNotificationConfiguration con un AWS SDK o una CLI
- Utilizzo PutBucketPolicy con un AWS SDK o una CLI
- Utilizzo PutBucketReplication con un AWS SDK o una CLI
- Utilizzo PutBucketRequestPayment con un AWS SDK o una CLI
- Utilizzo PutBucketTagging con un AWS SDK o una CLI
- Utilizzo PutBucketVersioning con un AWS SDK o una CLI
- Utilizzo PutBucketWebsite con un AWS SDK o una CLI
- Utilizzo PutObject con un AWS SDK o una CLI
- Utilizzo PutObjectAcl con un AWS SDK o una CLI
- Utilizzo PutObjectLegalHold con un AWS SDK o una CLI
- Utilizzo PutObjectLockConfiguration con un AWS SDK o una CLI
- Utilizzo PutObjectRetention con un AWS SDK o una CLI
- Utilizzo RestoreObject con un AWS SDK o una CLI
- Utilizzo SelectObjectContent con un AWS SDK o una CLI
- Utilizzo UploadPart con un AWS SDK o una CLI

Utilizzo **AbortMultipartUpload** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`AbortMultipartUpload`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Elimina caricamenti multiparte incompleti

CLI

AWS CLI

Per interrompere il caricamento multiparte specificato

Il `abort-multipart-upload` comando seguente interrompe un caricamento in più parti per la chiave `multipart/01` nel bucket. `my-bucket`

```
aws s3api abort-multipart-upload \
    --bucket my-bucket \
    --key multipart/01 \
    --upload-id
 dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3U
```

L'ID di caricamento richiesto da questo comando viene emesso da `create-multipart-upload` e può essere recuperato anche con. `list-multipart-uploads`

- Per i dettagli sull'API, consulta AbortMultipartUpload AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando interrompe i caricamenti in più parti creati prima di 5 giorni fa.

```
Remove-S3MultipartUpload -BucketName test-files -DaysBefore 5
```

Esempio 2: questo comando interrompe i caricamenti in più parti creati prima del 2 gennaio 2014.

```
Remove-S3MultipartUpload -BucketName test-files -InitiatedDate "Thursday, January
 02, 2014"
```

Esempio 3: questo comando interrompe i caricamenti in più parti creati prima del 2 gennaio 2014, 10:45:37.

```
Remove-S3MultipartUpload -BucketName test-files -InitiatedDate "2014/01/02
 10:45:37"
```

- Per i dettagli sull'API, vedere in Cmdlet Reference. AbortMultipartUploadAWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`AbortMultipartUploads`** con un AWS SDK o una CLI

Il seguente esempio di codice mostra come utilizzare`AbortMultipartUploads`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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}");
            }
        }
    }
```

- Per i dettagli sull'API, AbortMultipartUploadsconsulta AWS SDK for .NET API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CompleteMultipartUpload`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CompleteMultipartUpload`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Eseguire una copia in più parti
- Esegui un caricamento in più parti
- Utilizzo dei checksum

CLI

AWS CLI

Il comando seguente completa un caricamento in più parti per la chiave `multipart/01` nel bucket: `my-bucket`

```
aws s3api complete-multipart-upload --multipart-upload file://
mpustruct --bucket my-bucket --key 'multipart/01' --upload-id
 dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3U
```

L'ID di caricamento richiesto da questo comando viene emesso da `create-multipart-upload` e può essere recuperato anche con. `list-multipart-uploads`

L'opzione di caricamento in più parti del comando precedente utilizza una struttura JSON che descrive le parti del caricamento in più parti che devono essere riassemblate nel file completo. In questo esempio, il `file://` prefisso viene utilizzato per caricare la struttura JSON da un file nella cartella locale denominata. `mpustruct`

mpustruct:

```
{
  "Parts": [
    {
      "ETag": "e868e0f4719e394144ef36531ee6824c",
```

```
      "PartNumber": 1
    },
    {
      "ETag": "6bb2b12753d66fe86da4998aa33fffb0",
      "PartNumber": 2
    },
    {
      "ETag": "d0a0112e841abec9c9ec83406f0159c8",
      "PartNumber": 3
    }
  ]
}
```

Il valore ETag per ogni parte è upload viene emesso ogni volta che si carica una parte utilizzando il `upload-part` comando e può anche essere recuperato chiamando `list-parts` o calcolato utilizzando il checksum MD5 di ciascuna parte.

Output:

```
{
    "ETag": "\"3944a9f7a4faab7f78788ff6210f63f0-3\"",
    "Bucket": "my-bucket",
    "Location": "https://my-bucket.s3.amazonaws.com/multipart%2F01",
    "Key": "multipart/01"
}
```

- Per i dettagli sull'API, consulta CompleteMultipartUploadCommand Reference.AWS CLI

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
```

- Per i dettagli sulle API, consulta la CompleteMultipartUploadguida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CopyObject`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CopyObject`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Nozioni di base su bucket e oggetti
- Nozioni di base sulla crittografia

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
        }
    }
```

- Per i dettagli sull'API, CopyObjectconsulta AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta CopyObject AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
}
```

- Per i dettagli sull'API, CopyObjectconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Il comando seguente copia un oggetto da `bucket-1` a`bucket-2`:

```
aws s3api copy-object --copy-source bucket-1/test.txt --key test.txt --bucket
 bucket-2
```

Output:

```
{
    "CopyObjectResult": {
        "LastModified": "2015-11-10T01:07:25.000Z",
        "ETag": "\"589c8b79c230a6ecd5a7e1d040a9a030\""
    },
    "VersionId": "YdnYvTCVDqRRFA.NFJjy36p0hxifMlkA"
}
```

- Per i dettagli sull'API, vedere CopyObjectin AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, CopyObjectconsulta AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Copia un oggetto usando un'interfaccia 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 "";
    }
}
```

Usa un S3 TransferManager per copiare un oggetto da un bucket all'altro. Visualizza il file completo ed esegui il test.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.services.s3.model.CopyObjectRequest;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedCopy;
import software.amazon.awssdk.transfer.s3.model.Copy;
import software.amazon.awssdk.transfer.s3.model.CopyRequest;

import java.util.UUID;
```

```
    public String copyObject(S3TransferManager transferManager, String
 bucketName,
            String key, String destinationBucket, String destinationKey) {
        CopyObjectRequest copyObjectRequest = CopyObjectRequest.builder()
                .sourceBucket(bucketName)
                .sourceKey(key)
                .destinationBucket(destinationBucket)
                .destinationKey(destinationKey)
                .build();

        CopyRequest copyRequest = CopyRequest.builder()
                .copyObjectRequest(copyObjectRequest)
                .build();

        Copy copy = transferManager.copy(copyRequest);

        CompletedCopy completedCopy = copy.completionFuture().join();
        return completedCopy.response().copyObjectResult().eTag();
    }
```

- Per i dettagli sull'API, consulta la sezione AWS SDK for Java 2.x API CopyObjectReference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Copia l'oggetto.

```
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);
  }
};
```

- Per i dettagli sull'API, CopyObjectconsulta AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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)
    }
}
```

- Per i dettagli sull'API, CopyObjectconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Copia semplice di un oggetto.

```
        $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.");
        }
```

- Per i dettagli sull'API, CopyObjectconsulta AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo comando copia l'oggetto "sample.txt" dal bucket «test-files» allo stesso bucket ma con una nuova chiave "sample-copy.txt».

```
Copy-S3Object -BucketName test-files -Key sample.txt -DestinationKey sample-
copy.txt
```

Esempio 2: Questo comando copia l'oggetto "sample.txt" dal bucket «test-files» al bucket «backup-files» con una chiave "sample-copy.txt».

```
Copy-S3Object -BucketName test-files -Key sample.txt -DestinationKey sample-
copy.txt -DestinationBucket backup-files
```

Esempio 3: Questo comando scarica l'oggetto "sample.txt" dal bucket «test-files» in un file locale con nome "local-sample.txt».

```
Copy-S3Object -BucketName test-files -Key sample.txt -LocalFile local-sample.txt
```

Esempio 4: scarica il singolo oggetto nel file specificato. Il file scaricato si trova in c:\downloads \data\archive.zip

```
Copy-S3Object -BucketName test-files -Key data/archive.zip -LocalFolder c:
\downloads
```

Esempio 5: scarica tutti gli oggetti che corrispondono al prefisso chiave specificato nella cartella locale. La gerarchia delle chiavi relativa verrà conservata come sottocartelle nella posizione generale di download.

```
Copy-S3Object -BucketName test-files -KeyPrefix data -LocalFolder c:\downloads
```

- Per i dettagli sull'API, vedere CopyObjectin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta CopyObject AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Copia un oggetto.

```
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__
```

Copia un oggetto e aggiungi la crittografia lato server all'oggetto di destinazione.

```
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__
```

- Per i dettagli sull'API, CopyObjectconsulta AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sulle API, consulta la CopyObjectguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.
```

- Per i dettagli sulle API, CopyObjectconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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)
    }
```

- Per i dettagli sull'API, consulta la CopyObjectguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateBucket`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateBucket`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Nozioni di base su bucket e oggetti
- Utilizzo degli oggetti con versione

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
            }
        }
```

Crea un bucket con il blocco degli oggetti abilitato.

```
    /// <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;
        }
    }
```

- Per i dettagli sull'API, consulta la sezione CreateBucket AWS SDK for .NETAPI Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta CreateBucket AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
}
```

- Per i dettagli sull'API, CreateBucketconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Esempio 1: creare un bucket

L'`create-bucket`esempio seguente crea un bucket denominato: `my-bucket`

```
aws s3api create-bucket \
```

```
    --bucket my-bucket \
    --region us-east-1
```

Output:

```
{
    "Location": "/my-bucket"
}
```

Per ulteriori informazioni, consulta Creare un bucket nella Guida per l'utente di Amazon S3.

Esempio 2: creare un bucket con il proprietario imposto

L'`create-bucket`esempio seguente crea un bucket denominato `my-bucket` che utilizza l'impostazione bucket owner enforced per S3 Object Ownership.

```
aws s3api create-bucket \
    --bucket my-bucket \
    --region us-east-1 \
    --object-ownership BucketOwnerEnforced
```

Output:

```
{
    "Location": "/my-bucket"
}
```

Per ulteriori informazioni, consulta Controlling ownership of objects and disabling ACLs (Controllo della proprietà degli oggetti e disabilitazione degli ACL) nella Guida per l'utente di Amazon S3.

Esempio 3: creare un bucket al di fuori della regione ``us-east-1``

L'`create-bucket`esempio seguente crea un bucket denominato nella regione. `my-bucket` `eu-west-1` Le aree esterne `LocationConstraint` a `us-east-1` richiedono che venga specificato il valore appropriato per creare il bucket nella regione desiderata.

```
aws s3api create-bucket \
    --bucket my-bucket \
    --region eu-west-1 \
```

```
    --create-bucket-configuration LocationConstraint=eu-west-1
```

Output:

```
{
    "Location": "http://my-bucket.s3.amazonaws.com/"
}
```

Per ulteriori informazioni, consulta Creare un bucket nella Guida per l'utente di Amazon S3.

- Per i dettagli sull'API, consulta AWS CLI Command CreateBucketReference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea un bucket con configurazione predefinita.

```
// 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
}
```

Crea un bucket con blocco degli oggetti e attendi che esista.

```
// 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
}
```

- Per i dettagli sull'API, consulta la sezione AWS SDK for Go API CreateBucketReference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Creare 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);
        }
    }
}
```

Crea un bucket con il blocco degli oggetti abilitato.

```
    // 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");
    }
```

- Per i dettagli sull'API, consulta la sezione CreateBucket AWS SDK for Java 2.xAPI Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea il 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);
  }
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, CreateBucketconsulta AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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")
    }
}
```

- Per i dettagli sull'API, CreateBucketconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Creare 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.");
        }
```

- Per i dettagli sull'API, CreateBucketconsulta AWS SDK for PHP API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea un bucket con le impostazioni di default.

```
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
```

Crea un bucket con versione con una configurazione del ciclo di vita.

```
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
```

- Per i dettagli sull'API, consulta CreateBucket AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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__
```

- Per i dettagli sull'API, CreateBucketconsulta AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sulle API, consulta la CreateBucketguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.
```

- Per i dettagli sulle API, CreateBucketconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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)
    }
```

- Per i dettagli sull'API, consulta la CreateBucketguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateMultiRegionAccessPoint`** con un AWS SDK o una CLI

Il seguente esempio di codice mostra come utilizzare`CreateMultiRegionAccessPoint`.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Configura il client di controllo S3 per inviare la richiesta alla regione 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
        }
```

Crea il punto di accesso multiregionale.

```
    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"
    }
```

Attendi che il punto di accesso multiregionale diventi disponibile.

```
        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)
            }
        }
```

- Per ulteriori informazioni, consulta la Guida per gli sviluppatori di AWS SDK per Swift.

- Per i dettagli sull'API, consulta il riferimento CreateMultiRegionAccessPointall'API AWS SDK for Kotlin.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateMultipartUpload`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateMultipartUpload`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Eseguire una copia in più parti
- Esegui un caricamento in più parti
- Utilizzo dei checksum

CLI

AWS CLI

Il comando seguente crea un caricamento in più parti nel bucket `my-bucket` con la chiave: `multipart/01`

```
aws s3api create-multipart-upload --bucket my-bucket --key 'multipart/01'
```

Output:

```
{
    "Bucket": "my-bucket",
    "UploadId":
 "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3
    "Key": "multipart/01"
}
```

Il file completato verrà denominato `01` in una cartella chiamata `multipart` nel bucket. `my-bucket` Salva l'ID di caricamento, la chiave e il nome del bucket da utilizzare con il `upload-part` comando.

- Per i dettagli sull'API, consulta CreateMultipartUpload AWS CLICommand Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
let multipart_upload_res: CreateMultipartUploadOutput = client
    .create_multipart_upload()
    .bucket(&bucket_name)
    .key(&key)
    .send()
    .await
    .unwrap();
```

- Per i dettagli sulle API, consulta la CreateMultipartUploadguida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteBucket`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteBucket`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base su bucket e oggetti

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
        }
```

- Per i dettagli sull'API, DeleteBucketconsulta AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta DeleteBucket AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
}
```

- Per i dettagli sull'API, DeleteBucketconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Il comando seguente elimina un bucket denominato: `my-bucket`

```
aws s3api delete-bucket --bucket my-bucket --region us-east-1
```

- Per i dettagli sull'API, vedere DeleteBucketin AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, DeleteBucketconsulta AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        DeleteBucketRequest deleteBucketRequest = DeleteBucketRequest.builder()
                .bucket(bucket)
                .build();

        s3.deleteBucket(deleteBucketRequest);
        s3.close();
```

- Per i dettagli sull'API, DeleteBucketconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina il 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);
  }
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, DeleteBucketconsulta AWS SDK for JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina un bucket vuoto.

```
        $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.");
        }
```

- Per i dettagli sull'API, DeleteBucketconsulta AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando rimuove tutti gli oggetti e le versioni degli oggetti dal bucket 'test-files', quindi elimina il bucket. Il comando richiederà una conferma prima di procedere. Aggiungere l'interruttore -Force per sopprimere la conferma. Nota che i bucket che non sono vuoti non possono essere eliminati.

```
Remove-S3Bucket -BucketName test-files -DeleteBucketContent
```

- Per i dettagli sull'API, vedere DeleteBucketin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta DeleteBucket AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, DeleteBucketconsulta AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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(())
}
```

- Per i dettagli sulle API, consulta la DeleteBucketguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.
```

- Per i dettagli sulle API, DeleteBucketconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    public func deleteBucket(name: String) async throws {
        let input = DeleteBucketInput(
            bucket: name
        )
        _ = try await client.deleteBucket(input: input)
    }
```

- Per i dettagli sull'API, consulta la DeleteBucketguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteBucketAnalyticsConfiguration`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteBucketAnalyticsConfiguration`.

CLI

AWS CLI

Per eliminare una configurazione di analisi per un bucket

L'`delete-bucket-analytics-configuration`esempio seguente rimuove la configurazione di analisi per il bucket e l'ID specificati.

```
aws s3api delete-bucket-analytics-configuration \
    --bucket my-bucket \
    --id 1
```

Questo comando non produce alcun output.

- Per i dettagli sull'API, consulta DeleteBucketAnalyticsConfiguration AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: il comando rimuove il filtro di analisi con il nome 'testfilter' nel bucket S3 specificato.

```
Remove-S3BucketAnalyticsConfiguration -BucketName 's3testbucket' -AnalyticsId
 'testfilter'
```

- Per i dettagli sull'API, vedere DeleteBucketAnalyticsConfigurationin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteBucketCors`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteBucketCors`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
```

- Per i dettagli sull'API, DeleteBucketCorsconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Il comando seguente elimina una configurazione Cross-Origin Resource Sharing da un bucket denominato: `my-bucket`

```
aws s3api delete-bucket-cors --bucket my-bucket
```

- Per i dettagli sull'API, vedere DeleteBucketCorsin AWS CLI Command Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta DeleteBucketCors AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, DeleteBucketCorsconsulta AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteBucketEncryption`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteBucketEncryption`.

CLI

AWS CLI

Per eliminare la configurazione di crittografia lato server di un bucket

L'`delete-bucket-encryption`esempio seguente elimina la configurazione di crittografia lato server del bucket specificato.

```
aws s3api delete-bucket-encryption \
    --bucket my-bucket
```

Questo comando non produce alcun output.

- Per i dettagli sull'API, vedere DeleteBucketEncryptionin Command Reference.AWS CLI

PowerShell

Strumenti per PowerShell

Esempio 1: questo disabilita la crittografia abilitata per il bucket S3 fornito.

```
Remove-S3BucketEncryption -BucketName 's3casetestbucket'
```

Output:

```
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
```

- Per i dettagli sull'API, vedere DeleteBucketEncryptionin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteBucketInventoryConfiguration`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteBucketInventoryConfiguration`.

CLI

AWS CLI

Per eliminare la configurazione di inventario di un bucket

L'`delete-bucket-inventory-configuration`esempio seguente elimina la configurazione dell'inventario con ID 1 per il bucket specificato.

```
aws s3api delete-bucket-inventory-configuration \
    --bucket my-bucket \
    --id 1
```

Questo comando non produce alcun output.

- Per i dettagli sull'API, vedere DeleteBucketInventoryConfigurationin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando rimuove l'inventario denominato 'testInventoryName' corrispondente al bucket S3 specificato.

```
Remove-S3BucketInventoryConfiguration -BucketName 's3testbucket' -InventoryId
 'testInventoryName'
```

Output:

```
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
```

- Per i dettagli sull'API, vedere DeleteBucketInventoryConfigurationin Cmdlet Reference.AWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteBucketLifecycle`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteBucketLifecycle`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
```

- Per i dettagli sull'API, DeleteBucketLifecycleconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Il comando seguente elimina una configurazione del ciclo di vita da un bucket denominato: `my-bucket`

```
aws s3api delete-bucket-lifecycle --bucket my-bucket
```

- Per i dettagli sull'API, vedere DeleteBucketLifecyclein Command Reference.AWS CLI

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta DeleteBucketLifecycle AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteBucketMetricsConfiguration`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteBucketMetricsConfiguration`.

CLI

AWS CLI

Per eliminare una configurazione di metriche per un bucket

L'`delete-bucket-metrics-configuration`esempio seguente rimuove la configurazione delle metriche per il bucket e l'ID specificati.

```
aws s3api delete-bucket-metrics-configuration \
    --bucket my-bucket \
    --id 123
```

Questo comando non produce alcun output.

- Per i dettagli sull'API, consultate AWS CLI Command DeleteBucketMetricsConfigurationReference.

PowerShell

Strumenti per PowerShell

Esempio 1: il comando rimuove il filtro delle metriche con il nome 'testmetrics' nel bucket S3 specificato.

```
Remove-S3BucketMetricsConfiguration -BucketName 's3testbucket' -MetricsId
 'testmetrics'
```

- Per i dettagli sull'API, vedere in Cmdlet Reference. DeleteBucketMetricsConfigurationAWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteBucketPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteBucketPolicy`.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
}
```

- Per i dettagli sull'API, DeleteBucketPolicyconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Il comando seguente elimina una policy sui bucket da un bucket denominato: `my-bucket`

```
aws s3api delete-bucket-policy --bucket my-bucket
```

- Per i dettagli sull'API, vedere DeleteBucketPolicyin AWS CLI Command Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
}
```

- Per i dettagli sull'API, DeleteBucketPolicyconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina la policy 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);
  }
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, DeleteBucketPolicyconsulta AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun deleteS3BucketPolicy(bucketName: String?) {
    val request =
        DeleteBucketPolicyRequest {
            bucket = bucketName
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.deleteBucketPolicy(request)
        println("Done!")
    }
}
```

- Per i dettagli sull'API, DeleteBucketPolicyconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: il comando rimuove la policy del bucket associata al bucket S3 specificato.

```
Remove-S3BucketPolicy -BucketName 's3testbucket'
```

- Per i dettagli sull'API, vedere DeleteBucketPolicyin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta DeleteBucketPolicy AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, DeleteBucketPolicyconsulta AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteBucketReplication`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteBucketReplication`.

CLI

AWS CLI

Il comando seguente elimina una configurazione di replica da un bucket denominato: `my-bucket`

```
aws s3api delete-bucket-replication --bucket my-bucket
```

- Per i dettagli sull'API, vedere DeleteBucketReplicationin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: elimina la configurazione di replica associata al bucket denominato 'mybucket'. Nota che questa operazione richiede l'autorizzazione per l'azione s3:. DeleteReplicationConfiguration Ti verrà richiesta una conferma prima che l'operazione prosegua. Per sopprimere la conferma, usa l'interruttore -Force.

```
Remove-S3BucketReplication -BucketName mybucket
```

- Per i dettagli sull'API, vedere DeleteBucketReplicationin Cmdlet Reference.AWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteBucketTagging`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteBucketTagging`.

CLI

AWS CLI

Il comando seguente elimina una configurazione di tag da un bucket denominato: `my-bucket`

```
aws s3api delete-bucket-tagging --bucket my-bucket
```

- Per i dettagli sull'API, vedere DeleteBucketTaggingin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando rimuove tutti i tag associati al bucket S3 specificato.

```
Remove-S3BucketTagging -BucketName 's3testbucket'
```

Output:

```
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
```

- Per i dettagli sull'API, vedere DeleteBucketTaggingin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteBucketWebsite`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteBucketWebsite`.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
}
```

- Per i dettagli sull'API, DeleteBucketWebsiteconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Il comando seguente elimina la configurazione di un sito Web da un bucket denominato: `my-bucket`

```
aws s3api delete-bucket-website --bucket my-bucket
```

- Per i dettagli sull'API, consulta AWS CLI Command DeleteBucketWebsiteReference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
}
```

- Per i dettagli sull'API, DeleteBucketWebsiteconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina la configurazione del sito Web dal 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);
  }
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, DeleteBucketWebsiteconsulta AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando disabilita la proprietà statica di hosting del sito Web del bucket S3 specificato.

```
Remove-S3BucketWebsite -BucketName 's3testbucket'
```

Output:

```
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
```

- Per i dettagli sull'API, vedere DeleteBucketWebsitein AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteObject`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteObject`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Utilizzo degli oggetti con versione

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina un oggetto in un bucket S3 senza controllo delle versioni.

```
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.");
            }
        }
    }
```

Elimina un oggetto in un bucket S3 con controllo delle versioni.

```
    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;
        }
    }
```

- Per i dettagli sull'API, DeleteObjectconsulta AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta DeleteObject AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
}
```

- Per i dettagli sull'API, DeleteObjectconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Il comando seguente elimina un oggetto denominato `test.txt` da un bucket denominato: `my-bucket`

```
aws s3api delete-object --bucket my-bucket --key test.txt
```

Se il controllo delle versioni del bucket è abilitato, l'output conterrà l'ID di versione del marker di eliminazione:

```
{
```

```
  "VersionId": "9_gKg5vG56F.TTEUdwkxGpJ3tNDlWlGq",
  "DeleteMarker": true
}
```

Per ulteriori informazioni sull'eliminazione di oggetti, consulta Deleting Objects nella Amazon S3 Developer Guide.

- Per i dettagli sull'API, consulta Command DeleteObjectReference AWS CLI .

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, DeleteObjectconsulta AWS SDK for Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina un oggetto.

```
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);
  }
};
```

- Per i dettagli sull'API, DeleteObjectconsulta AWS SDK for JavaScript API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina un oggetto.

```
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
```

Ripristina la versione precedente di un oggetto eliminando quelle successive.

```
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}."
        )
```

Riattiva un oggetto eliminato rimuovendo il contrassegno di eliminazione attivo dell'oggetto.

```
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)
```

Crea un gestore Lambda che rimuove un contrassegno di eliminazione da un oggetto S3. Questo gestore può essere utilizzato per ripulire in modo efficiente i contrassegni di eliminazione estranei di un bucket con versione.

```
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,
    }
```

- Per i dettagli sull'API, consulta DeleteObject AWSSDK for Python (Boto3) API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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(())
}
```

- Per i dettagli sulle API, consulta la DeleteObjectguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.
```

- Per i dettagli sulle API, DeleteObjectconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
    }
}
```

- Per i dettagli sull'API, consulta la DeleteObjectguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteObjectTagging`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteObjectTagging`.

CLI

AWS CLI

Per eliminare i set di tag di un oggetto

L'`delete-object-tagging`esempio seguente elimina il tag con la chiave specificata dall'oggetto`doc1.rtf`.

```
aws s3api delete-object-tagging \
    --bucket my-bucket \
    --key doc1.rtf
```

Questo comando non produce alcun output.

- Per i dettagli sull'API, consultate DeleteObjectTagging AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando rimuove tutti i tag associati all'oggetto con la chiave 'testfile.txt' nel bucket S3 specificato.

```
Remove-S3ObjectTagSet -Key 'testfile.txt' -BucketName 's3testbucket' -Select
 '^Key'
```

Output:

```
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
```

- Per i dettagli sull'API, vedere DeleteObjectTaggingin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteObjects`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteObjects`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base su bucket e oggetti

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Eliminazione di tutti gli oggetti da un bucket Amazon 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;
            }
        }
```

Eliminare più oggetti da un bucket S3 senza controllo delle versioni.

```
   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;
        }
    }
```

Eliminare più oggetti da un bucket S3 con controllo delle versioni.

```
    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;
        }
    }
```

- Per i dettagli sull'API, DeleteObjectsconsulta AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta DeleteObjects AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
}
```

- Per i dettagli sull'API, DeleteObjectsconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Il comando seguente elimina un oggetto da un bucket denominato: `my-bucket`

```
aws s3api delete-objects --bucket my-bucket --delete file://delete.json
```

`delete.json`è un documento JSON nella directory corrente che specifica l'oggetto da eliminare:

```
{
  "Objects": [
```

```
    {
      "Key": "test1.txt"
    }
  ],
  "Quiet": false
}
```

Output:

```
{
    "Deleted": [
        {
            "DeleteMarkerVersionId": "mYAT5Mc6F7aeUL8SS7FAAqUPO1koHwzU",
            "Key": "test1.txt",
            "DeleteMarker": true
        }
    ]
}
```

- Per i dettagli sull'API, consulta AWS CLI Command DeleteObjectsReference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, DeleteObjectsconsulta AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
}
```

- Per i dettagli sull'API, DeleteObjectsconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina più oggetti.

```
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);
  }
};
```

- Per i dettagli sull'API, DeleteObjectsconsulta AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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")
    }
}
```

- Per i dettagli sull'API, DeleteObjectsconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina un set di oggetti da un elenco di chiavi.

```
        $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.");
        }
```

- Per i dettagli sull'API, DeleteObjectsconsulta AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo comando rimuove l'oggetto "sample.txt" dal bucket «test-files». Viene richiesta una conferma prima dell'esecuzione del comando; per sopprimere il prompt, utilizzare l'opzione -Force.

```
Remove-S3Object -BucketName test-files -Key sample.txt
```

Esempio 2: questo comando rimuove la versione specificata dell'oggetto "sample.txt" dal bucket «test-files», presupponendo che il bucket sia stato configurato per abilitare le versioni degli oggetti.

```
Remove-S3Object -BucketName test-files -Key sample.txt -VersionId
 HLbxnx6V9omT6AQYVpks8mmFKQcejpqt
```

Esempio 3: Questo comando rimuove gli oggetti "sample1.txt «," sample2.txt "e" sample3.txt "dal bucket «test-files» come un'unica operazione batch. La risposta del servizio elencherà tutte le chiavi elaborate, indipendentemente dallo stato di successo o di errore dell'eliminazione. Per ottenere solo gli errori relativi alle chiavi che non hanno potuto essere elaborate dal servizio, aggiungi il ReportErrorsOnly parametro - (questo parametro può essere specificato anche con l'alias -Quiet).

```
Remove-S3Object -BucketName test-files -KeyCollection @( "sample1.txt",
 "sample2.txt", "sample3.txt" )
```

Esempio 4: Questo esempio utilizza un'espressione in linea con il KeyCollection parametro - per ottenere le chiavi degli oggetti da eliminare. Get-S3Objectrestituisce una raccolta di istanze Amazon.S3.Model.S3Object, ognuna delle quali ha un membro Key di tipo string che identifica l'oggetto.

```
Remove-S3Object -bucketname "test-files" -KeyCollection (Get-S3Object "test-
files" -KeyPrefix "prefix/subprefix" | select -ExpandProperty Key)
```

Esempio 5: Questo esempio ottiene tutti gli oggetti che hanno un prefisso chiave «prefix/subprefix» nel bucket e li elimina. Si noti che gli oggetti in entrata vengono elaborati uno alla volta. Per raccolte di grandi dimensioni, è consigliabile passare la raccolta al parametro - InputObject (alias -S3ObjectCollection) del cmdlet per consentire l'eliminazione come batch con una singola chiamata al servizio.

```
Get-S3Object -BucketName "test-files" -KeyPrefix "prefix/subprefix" | Remove-
S3Object -Force
```

Esempio 6: questo esempio reindirizza una raccolta di istanze Amazon.S3.Model.S3 ObjectVersion che rappresentano indicatori di eliminazione al cmdlet per l'eliminazione. Tieni presente che gli oggetti in entrata vengono elaborati uno alla volta. Per raccolte di grandi dimensioni, è consigliabile passare la raccolta al parametro - InputObject (alias - S3ObjectCollection) del cmdlet per consentire l'eliminazione come batch con una singola chiamata al servizio.

```
(Get-S3Version -BucketName "test-files").Versions | Where {$_.IsDeleteMarker -eq
 "True"} | Remove-S3Object -Force
```

Esempio 7: questo script mostra come eseguire un'eliminazione in batch di un set di oggetti (in questo caso eliminare i marker) creando una matrice di oggetti da utilizzare con il parametro -. 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
```

- Per i dettagli sull'API, vedere DeleteObjectsin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina un set di oggetti utilizzando un elenco di chiavi oggetto.

```
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
```

Elimina tutti gli oggetti in 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
```

Elimina in modo permanente un oggetto con versione eliminando tutte le relative versioni.

```
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
```

- Per i dettagli sull'API, consulta DeleteObjects AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, DeleteObjectsconsulta AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.",
        )),
    }
}
```

- Per i dettagli sulle API, consulta la DeleteObjectsguida di riferimento all'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
        }
    }
```

- Per i dettagli sull'API, consulta la DeleteObjectsguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeletePublicAccessBlock`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeletePublicAccessBlock`.

CLI

AWS CLI

Per eliminare la configurazione di blocco dell'accesso pubblico per un bucket

L'`delete-public-access-block`esempio seguente rimuove la configurazione di accesso pubblico a blocchi sul bucket specificato.

```
aws s3api delete-public-access-block \
    --bucket my-bucket
```

Questo comando non produce alcun output.

- Per i dettagli sull'API, vedere DeletePublicAccessBlockin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando disattiva l'impostazione di blocco dell'accesso pubblico per il bucket specificato.

```
Remove-S3PublicAccessBlock -BucketName 's3testbucket' -Force -Select
 '^BucketName'
```

Output:

```
s3testbucket
```

- Per i dettagli sull'API, vedere DeletePublicAccessBlockin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **GetBucketAccelerateConfiguration** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketAccelerateConfiguration`.

CLI

AWS CLI

Per recuperare la configurazione accelerata di un bucket

L'`get-bucket-accelerate-configuration`esempio seguente recupera la configurazione di accelerazione per il bucket specificato.

```
aws s3api get-bucket-accelerate-configuration \
    --bucket my-bucket
```

Output:

```
{
    "Status": "Enabled"
}
```

- Per i dettagli sull'API, vedere GetBucketAccelerateConfigurationin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce il valore Enabled, se le impostazioni di accelerazione del trasferimento sono abilitate per il bucket specificato.

```
Get-S3BucketAccelerateConfiguration -BucketName 's3testbucket'
```

Output:

```
Value
-----
Enabled
```

- Per i dettagli sull'API, vedere GetBucketAccelerateConfigurationin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetBucketAcl`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketAcl`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestire le liste di controllo degli accessi (ACL)

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
        }
```

- Per i dettagli sull'API, GetBucketAclconsulta AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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";
}
```

- Per i dettagli sull'API, GetBucketAclconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Il comando seguente recupera l'elenco di controllo degli accessi per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-acl --bucket my-bucket
```

Output:

```
{
    "Owner": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
            },
            "Permission": "FULL_CONTROL"
        }
    ]
}
```

- Per i dettagli sull'API, vedere GetBucketAclin AWS CLI Command Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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 "";
    }
}
```

- Per i dettagli sull'API, GetBucketAclconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Recupera le autorizzazioni 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);
  }
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, GetBucketAclconsulta AWS SDK for JavaScript API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta GetBucketAcl AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **GetBucketAnalyticsConfiguration** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketAnalyticsConfiguration`.

CLI

AWS CLI

Per recuperare la configurazione di analisi per un bucket con un ID specifico

L'`get-bucket-analytics-configuration`esempio seguente mostra la configurazione di analisi per il bucket e l'ID specificati.

```
aws s3api get-bucket-analytics-configuration \
    --bucket my-bucket \
    --id 1
```

Output:

```
{
    "AnalyticsConfiguration": {
        "StorageClassAnalysis": {},
        "Id": "1"
    }
}
```

- Per i dettagli sull'API, consulta GetBucketAnalyticsConfiguration AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce i dettagli del filtro di analisi con il nome 'testfilter' nel bucket S3 specificato.

```
Get-S3BucketAnalyticsConfiguration -BucketName 's3testbucket' -AnalyticsId
 'testfilter'
```

- Per i dettagli sull'API, vedere GetBucketAnalyticsConfigurationin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetBucketCors`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketCors`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
        }
```

- Per i dettagli sull'API, GetBucketCorsconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Il comando seguente recupera la configurazione Cross-Origin Resource Sharing per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-cors --bucket my-bucket
```

Output:

```
{
    "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": [
                "*"
            ]
        }
    ]
}
```

- Per i dettagli sull'API, vedere GetBucketCorsin AWS CLI Command Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Recupera la policy CORS per il 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);
  }
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, GetBucketCorsconsulta AWS SDK for JavaScript API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta GetBucketCors AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, GetBucketCorsconsulta AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetBucketEncryption`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketEncryption`.

CLI

AWS CLI

Per recuperare la configurazione di crittografia lato server per un bucket

L'`get-bucket-encryption`esempio seguente recupera la configurazione di crittografia lato server per il bucket. `my-bucket`

```
aws s3api get-bucket-encryption \
```

```
    --bucket my-bucket
```

Output:

```
{
    "ServerSideEncryptionConfiguration": {
        "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "AES256"
                }
            }
        ]
    }
}
```

- Per i dettagli sull'API, vedere GetBucketEncryptionin Command Reference.AWS CLI

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce tutte le regole di crittografia lato server associate al bucket specificato.

```
Get-S3BucketEncryption -BucketName 's3casetestbucket'
```

- Per i dettagli sull'API, vedere GetBucketEncryptionin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetBucketInventoryConfiguration`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketInventoryConfiguration`.

CLI

AWS CLI

Per recuperare la configurazione di inventario per un bucket

L'`get-bucket-inventory-configuration`esempio seguente recupera la configurazione dell'inventario per il bucket specificato con ID. 1

```
aws s3api get-bucket-inventory-configuration \
    --bucket my-bucket \
    --id 1
```

Output:

```
{
    "InventoryConfiguration": {
        "IsEnabled": true,
        "Destination": {
            "S3BucketDestination": {
                "Format": "ORC",
                "Bucket": "arn:aws:s3:::my-bucket",
                "AccountId": "123456789012"
            }
        },
        "IncludedObjectVersions": "Current",
        "Id": "1",
        "Schedule": {
            "Frequency": "Weekly"
        }
    }
}
```

- Per i dettagli sull'API, vedere GetBucketInventoryConfigurationin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce i dettagli dell'inventario denominato 'testinventory' per il bucket S3 specificato.

```
Get-S3BucketInventoryConfiguration -BucketName 's3testbucket' -InventoryId
 'testinventory'
```

- Per i dettagli sull'API, vedere GetBucketInventoryConfigurationin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **GetBucketLifecycleConfiguration** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketLifecycleConfiguration`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
        }
```

- Per i dettagli sull'API, GetBucketLifecycleConfigurationconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Il comando seguente recupera la configurazione del ciclo di vita per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-lifecycle-configuration --bucket my-bucket
```

Output:

```
{
    "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"
        }
    ]
}
```

- Per i dettagli sull'API, vedere GetBucketLifecycleConfigurationin Command Reference.AWS CLI

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta GetBucketLifecycleConfiguration AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetBucketLocation`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketLocation`.

CLI

AWS CLI

Il comando seguente recupera il vincolo di posizione per un bucket denominato`my-bucket`, se esiste un vincolo:

```
aws s3api get-bucket-location --bucket my-bucket
```

Output:

```
{
    "LocationConstraint": "us-west-2"
```

```
}
```

- Per i dettagli sull'API, consulta Command Reference. GetBucketLocationAWS CLI

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce il vincolo di posizione per il bucket 's3testbucket', se esiste un vincolo.

```
Get-S3BucketLocation -BucketName 's3testbucket'
```

Output:

```
Value
-----
ap-south-1
```

- Per i dettagli sull'API, vedere in Cmdlet Reference. GetBucketLocationAWS Tools for PowerShell

Rust

SDK per Rust

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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(())
}
```

- Per i dettagli sulle API, consulta la GetBucketLocationguida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetBucketLogging`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketLogging`.

CLI

AWS CLI

Per recuperare lo stato di registrazione di un bucket

L'`get-bucket-logging`esempio seguente recupera lo stato di registrazione per il bucket specificato.

```
aws s3api get-bucket-logging \
    --bucket my-bucket
```

Output:

```
{
    "LoggingEnabled": {
        "TargetPrefix": "",
        "TargetBucket": "my-bucket-logs"
          }
}
```

- Per i dettagli sull'API, vedere GetBucketLoggingin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce lo stato di registrazione per il bucket specificato.

```
Get-S3BucketLogging -BucketName 's3testbucket'
```

Output:

```
TargetBucketName   Grants TargetPrefix
----------------   ------ ------------
testbucket1        {}     testprefix
```

- Per i dettagli sull'API, vedere GetBucketLoggingin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **GetBucketMetricsConfiguration** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketMetricsConfiguration`.

CLI

AWS CLI

Per recuperare la configurazione delle metriche per un bucket con un ID specifico

L'`get-bucket-metrics-configuration`esempio seguente visualizza la configurazione delle metriche per il bucket e l'ID specificati.

```
aws s3api get-bucket-metrics-configuration \
    --bucket my-bucket \
    --id 123
```

Output:

```
{
    "MetricsConfiguration": {
        "Filter": {
            "Prefix": "logs"
        },
        "Id": "123"
    }
}
```

- Per i dettagli sull'API, consulta AWS CLI Command GetBucketMetricsConfigurationReference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce i dettagli sul filtro metrico denominato 'testfilter' per il bucket S3 specificato.

```
Get-S3BucketMetricsConfiguration -BucketName 's3testbucket' -MetricsId
 'testfilter'
```

- Per i dettagli sull'API, vedere GetBucketMetricsConfigurationin Cmdlet Reference.AWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetBucketNotification`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketNotification`.

CLI

AWS CLI

Il comando seguente recupera la configurazione di notifica per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-notification --bucket my-bucket
```

Output:

```
{
    "TopicConfiguration": {
        "Topic": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
        "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNl",
        "Event": "s3:ObjectCreated:*",
        "Events": [
            "s3:ObjectCreated:*"
        ]
    }
}
```

- Per i dettagli sull'API, vedere GetBucketNotificationin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo esempio recupera la configurazione di notifica del bucket specificato

```
Get-S3BucketNotification -BucketName kt-tools | select -ExpandProperty
 TopicConfigurations
```

Output:

```
Id   Topic
--   -----
mimo arn:aws:sns:eu-west-1:123456789012:topic-1
```

- Per i dettagli sull'API, vedere GetBucketNotificationin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetBucketPolicy`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketPolicy`.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
}
```

- Per i dettagli sull'API, GetBucketPolicyconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Il comando seguente recupera la policy del bucket per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-policy --bucket my-bucket
```

Output:

```
{
```

```
    "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/*
\"}]}"
}
```

Ottieni e inserisci una bucket policyIl seguente esempio mostra come scaricare una policy sui bucket di Amazon S3, apportare modifiche al file e quindi `put-bucket-policy` utilizzarla per applicare la policy del bucket modificata. Per scaricare la bucket policy in un file, puoi eseguire:

```
aws s3api get-bucket-policy --bucket mybucket --query Policy --output text >
 policy.json
```

È quindi possibile modificare il `policy.json` file in base alle esigenze. Infine puoi riapplicare questa policy modificata al bucket S3 eseguendo:

`policy.json`file se necessario. Infine puoi riapplicare questa policy modificata al bucket S3 eseguendo:

file se necessario. Infine puoi riapplicare questa policy modificata al bucket S3 eseguendo:

```
aws s3api put-bucket-policy --bucket mybucket --policy file://policy.json
```

- Per i dettagli sull'API, consulta AWS CLI Command GetBucketPolicyReference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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 "";
    }
}
```

- Per i dettagli sull'API, GetBucketPolicyconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Recupera la policy 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);
  }
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, GetBucketPolicyconsulta AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
    }
}
```

- Per i dettagli sull'API, GetBucketPolicyconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce la policy del bucket associata al bucket S3 specificato.

```
Get-S3BucketPolicy -BucketName 's3testbucket'
```

- Per i dettagli sull'API, vedere GetBucketPolicyin Cmdlet Reference.AWS Tools for PowerShell

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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)
```

- Per i dettagli sull'API, consulta GetBucketPolicy AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, GetBucketPolicyconsulta AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetBucketPolicyStatus`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketPolicyStatus`.

CLI

AWS CLI

Per recuperare lo stato della politica di un bucket che indica se il bucket è pubblico

L'`get-bucket-policy-status`esempio seguente recupera lo stato della politica per il bucket. `my-bucket`

```
aws s3api get-bucket-policy-status \
    --bucket my-bucket
```

Output:

```
{
    "PolicyStatus": {
        "IsPublic": false
    }
}
```

- Per i dettagli sull'API, vedere GetBucketPolicyStatusin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce lo stato della politica per il bucket S3 specificato, indicando se il bucket è pubblico.

```
Get-S3BucketPolicyStatus -BucketName 's3casetestbucket'
```

- Per i dettagli sull'API, vedere GetBucketPolicyStatusin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetBucketReplication`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketReplication`.

CLI

AWS CLI

Il comando seguente recupera la configurazione di replica per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-replication --bucket my-bucket
```

Output:

```
{
    "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"
    }
}
```

- Per i dettagli sull'API, vedere GetBucketReplicationin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: restituisce le informazioni di configurazione della replica impostate nel bucket denominato 'mybucket'.

```
Get-S3BucketReplication -BucketName mybucket
```

- Per i dettagli sull'API, vedere in Cmdlet Reference. GetBucketReplicationAWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetBucketRequestPayment`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketRequestPayment`.

CLI

AWS CLI

Per recuperare la configurazione di pagamento della richiesta per un bucket

L'`get-bucket-request-payment`esempio seguente recupera la configurazione requester pays per il bucket specificato.

```
aws s3api get-bucket-request-payment \
```

```
    --bucket my-bucket
```

Output:

```
{
    "Payer": "BucketOwner"
}
```

- Per i dettagli sull'API, vedere GetBucketRequestPaymentin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: restituisce la configurazione di pagamento della richiesta per il bucket denominato 'mybucket'. Per impostazione predefinita, il proprietario del bucket paga per i download dal bucket.

```
Get-S3BucketRequestPayment -BucketName mybucket
```

- Per i dettagli sull'API, vedere GetBucketRequestPaymentin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetBucketTagging`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketTagging`.

CLI

AWS CLI

Il comando seguente recupera la configurazione dei tag per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-tagging --bucket my-bucket
```

Output:

```
{
    "TagSet": [
        {
            "Value": "marketing",
            "Key": "organization"
        }
    ]
}
```

- Per i dettagli sull'API, vedere GetBucketTaggingin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce tutti i tag associati al bucket specificato.

```
Get-S3BucketTagging -BucketName 's3casetestbucket'
```

- Per i dettagli sull'API, vedere GetBucketTaggingin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetBucketVersioning`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketVersioning`.

CLI

AWS CLI

Il comando seguente recupera la configurazione del controllo delle versioni per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-versioning --bucket my-bucket
```

Output:

```
{
    "Status": "Enabled"
}
```

- Per i dettagli sull'API, vedere GetBucketVersioningin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce lo stato del controllo delle versioni rispetto al bucket specificato.

```
Get-S3BucketVersioning -BucketName 's3testbucket'
```

- Per i dettagli sull'API, vedere GetBucketVersioningin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetBucketWebsite`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetBucketWebsite`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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}");
```

- Per i dettagli sull'API, GetBucketWebsiteconsulta AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
}
```

- Per i dettagli sull'API, GetBucketWebsiteconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Il comando seguente recupera la configurazione statica del sito Web per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-website --bucket my-bucket
```

Output:

```
{
    "IndexDocument": {
        "Suffix": "index.html"
    },
    "ErrorDocument": {
        "Key": "error.html"
    }
}
```

- Per i dettagli sull'API, consulta AWS CLI Command GetBucketWebsiteReference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Recupera la configurazione del sito 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);
  }
};
```

- Per i dettagli sull'API, GetBucketWebsiteconsulta AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce i dettagli delle configurazioni statiche del sito Web del bucket S3 specificato.

```
Get-S3BucketWebsite -BucketName 's3testbucket'
```

- Per i dettagli sull'API, vedere GetBucketWebsitein AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetObject`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetObject`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Recuperare un oggetto da un bucket se è stato modificato
- Ottieni un oggetto da un punto di accesso multiregionale
- Nozioni di base su bucket e oggetti
- Nozioni di base sulla crittografia
- Tieni traccia dei caricamenti e dei download

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
            }
        }
```

- Per i dettagli sull'API, GetObjectconsulta AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta GetObject AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
}
```

- Per i dettagli sull'API, GetObjectconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

L'esempio seguente utilizza il `get-object` comando per scaricare un oggetto da Amazon S3:

```
aws s3api get-object --bucket text-content --key dir/my_images.tar.bz2
 my_images.tar.bz2
```

Tieni presente che il parametro outfile è specificato senza un nome di opzione come «--outfile». Il nome del file di output deve essere l'ultimo parametro del comando.

L'esempio seguente dimostra l'utilizzo di `--range` per scaricare un intervallo di byte specifico da un oggetto. Nota che gli intervalli di byte devono essere preceduti dal prefisso «bytes=»:

```
aws s3api get-object --bucket text-content --key dir/my_data --range
 bytes=8888-9999 my_data_range
```

Per ulteriori informazioni sul recupero di oggetti, consulta Getting Objects nella Amazon S3 Developer Guide.

- Per i dettagli sull'API, consulta Command GetObjectReference AWS CLI .

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, GetObjectconsulta AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Leggi i dati come array di byte utilizzando un'interfaccia S3Client.

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
```

```
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class GetObjectData {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <bucketName> <keyName> <path>

                Where:
                    bucketName - The Amazon S3 bucket name.\s
                    keyName - The key name.\s
                    path - The path where the file is written to.\s
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String keyName = args[1];
        String path = args[2];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        getObjectBytes(s3, bucketName, keyName, path);
    }

    public static void getObjectBytes(S3Client s3, String bucketName, String
 keyName, String path) {
        try {
            GetObjectRequest objectRequest = GetObjectRequest
                    .builder()
                    .key(keyName)
                    .bucket(bucketName)
                    .build();

            ResponseBytes<GetObjectResponse> objectBytes =
 s3.getObjectAsBytes(objectRequest);
```

```
            byte[] data = objectBytes.asByteArray();

            // Write the data to a local file.
            File myFile = new File(path);
            OutputStream os = new FileOutputStream(myFile);
            os.write(data);
            System.out.println("Successfully obtained bytes from an S3 object");
            os.close();

        } catch (IOException ex) {
            ex.printStackTrace();
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

Usa un S3 TransferManager per scaricare un oggetto in un bucket S3 in un file locale. Visualizza il file completo ed esegui il test.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedFileDownload;
import software.amazon.awssdk.transfer.s3.model.DownloadFileRequest;
import software.amazon.awssdk.transfer.s3.model.FileDownload;
import software.amazon.awssdk.transfer.s3.progress.LoggingTransferListener;

import java.io.IOException;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.UUID;

    public Long downloadFile(S3TransferManager transferManager, String
 bucketName,
                             String key, String downloadedFileWithPath) {
```

```
        DownloadFileRequest downloadFileRequest = DownloadFileRequest.builder()
                .getObjectRequest(b -> b.bucket(bucketName).key(key))
                .destination(Paths.get(downloadedFileWithPath))
                .build();

        FileDownload downloadFile =
 transferManager.downloadFile(downloadFileRequest);

        CompletedFileDownload downloadResult =
 downloadFile.completionFuture().join();
        logger.info("Content length [{}]",
 downloadResult.response().contentLength());
        return downloadResult.response().contentLength();
    }
```

Leggi i tag appartenenti a un oggetto utilizzando un'interfaccia 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);
        }
    }
}
```

Recupera un URL per un oggetto utilizzando un'interfaccia 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);
        }
    }
}
```

Recupera un oggetto utilizzando l'oggetto client S3Presigner mediante un'interfaccia 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();
        }
    }
}
```

Ottieni un oggetto usando un ResponseTransformer oggetto e 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);
        }
    }
}
```

- Per i dettagli sull'API, vedi GetObjectin AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Scarica l'oggetto.

```
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);
  }
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, GetObjectconsulta AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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")
        }
    }
}
```

- Per i dettagli sull'API, GetObjectconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Recupera un oggetto.

```
        $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.");
        }
```

- Per i dettagli sull'API, GetObjectconsulta AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Questo comando recupera l'elemento "sample.txt" dal bucket «test-files» e lo salva in un file denominato "local-sample.txt" nella posizione corrente. Il file "local-sample.txt" non deve esistere prima che questo comando venga chiamato.

```
Read-S3Object -BucketName test-files -Key sample.txt -File local-sample.txt
```

Esempio 2: Questo comando recupera la directory virtuale «DIR» dal bucket «test-files» e la salva in una cartella denominata «Local-dir» nella posizione corrente. La cartella «Local-dir» non deve esistere prima che questo comando venga chiamato.

```
Read-S3Object -BucketName test-files -KeyPrefix DIR -Folder Local-DIR
```

Esempio 3: scarica tutti gli oggetti con chiavi che terminano con '.json' dai bucket con 'config' nel nome del bucket ai file nella cartella specificata. Le chiavi degli oggetti vengono utilizzate per impostare i nomi dei file.

```
Get-S3Bucket | ? { $_.BucketName -like '*config*' } | Get-S3Object | ? { $_.Key -
like '*.json' } | Read-S3Object -Folder C:\ConfigObjects
```

- Per i dettagli sull'API, vedere GetObjectin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta GetObject AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Recupera un oggetto.

```
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__
```

Recupera un oggetto e segnala lo stato di crittografia lato server.

```
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__
```

- Per i dettagli sull'API, GetObjectconsulta AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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)
}
```

- Per i dettagli sulle API, consulta la GetObjectguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.
```

- Per i dettagli sulle API, GetObjectconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Scaricamento di un oggetto da un bucket in un file locale.

```
   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)
    }
```

Lettura di un oggetto in un oggetto Swift Data.

```
    public func readFile(bucket: String, key: String) async throws -> Data {
        let input = GetObjectInput(
            bucket: bucket,
            key: key
        )
        let output = try await client.getObject(input: input)

        // Get the stream and return its contents in a `Data` object. If
        // there is no stream, return an empty `Data` object instead.
        guard let body = output.body,
              let data = try await body.readData() else {
            return "".data(using: .utf8)!
        }
```

```
        return data
    }
```

- Per i dettagli sull'API, consulta la GetObjectguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetObjectAcl`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetObjectAcl`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. Puoi vedere questa azione nel contesto nel seguente esempio di codice:

- Gestire le liste di controllo degli accessi (ACL)

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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";
    }
}
```

- Per i dettagli sull'API, GetObjectAclconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Il comando seguente recupera l'elenco di controllo degli accessi per un oggetto in un bucket denominato: `my-bucket`

```
aws s3api get-object-acl --bucket my-bucket --key index.html
```

Output:

```
{
    "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"
        }
    ]
}
```

- Per i dettagli sull'API, vedere GetObjectAclin AWS CLI Command Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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}")
        }
    }
}
```

- Per i dettagli sull'API, GetObjectAclconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta GetObjectAcl AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetObjectLegalHold`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetObjectLegalHold`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Ottieni la configurazione di conservazione legale di un oggetto
- Blocca oggetti Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
        }
    }
```

- Per i dettagli sull'API, GetObjectLegalHoldconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Recupera lo stato di conservazione legale di un oggetto

L'`get-object-legal-hold`esempio seguente recupera lo stato Legal Hold per l'oggetto specificato.

```
aws s3api get-object-legal-hold \
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf
```

Output:

```
{
    "LegalHold": {
        "Status": "ON"
    }
}
```

- Per i dettagli sull'API, consultate GetObjectLegalHold AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, GetObjectLegalHoldconsulta AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
    }
```

- Per i dettagli sull'API, GetObjectLegalHold consulta AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetObjectLockConfiguration`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetObjectLockConfiguration`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Blocca oggetti Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
        }
    }
```

- Per i dettagli sull'API, GetObjectLockConfigurationconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Per recuperare una configurazione di blocco degli oggetti per un bucket

L'`get-object-lock-configuration`esempio seguente recupera la configurazione del blocco degli oggetti per il bucket specificato.

```
aws s3api get-object-lock-configuration \
    --bucket my-bucket-with-object-lock
```

Output:

```
{
    "ObjectLockConfiguration": {
        "ObjectLockEnabled": "Enabled",
        "Rule": {
            "DefaultRetention": {
                "Mode": "COMPLIANCE",
                "Days": 50
            }
        }
    }
}
```

- Per i dettagli sull'API, vedere GetObjectLockConfigurationin AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, GetObjectLockConfigurationconsulta AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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());
    }
```

- Per i dettagli sull'API, GetObjectLockConfigurationconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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");
}
```

- Per i dettagli sull'API, GetObjectLockConfigurationconsulta AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce il valore 'Enabled' se la configurazione Object lock è abilitata per il bucket S3 specificato.

```
Get-S3ObjectLockConfiguration -BucketName 's3buckettesting' -Select
 ObjectLockConfiguration.ObjectLockEnabled
```

Output:

```
Value
-----
Enabled
```

- Per i dettagli sull'API, vedere GetObjectLockConfigurationin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetObjectRetention`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetObjectRetention`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Blocca oggetti Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
        }
    }
```

- Per i dettagli sull'API, GetObjectRetentionconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Per recuperare la configurazione di conservazione degli oggetti per un oggetto

L'`get-object-retention`esempio seguente recupera la configurazione di conservazione degli oggetti per l'oggetto specificato.

```
aws s3api get-object-retention \
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf
```

Output:

```
{
    "Retention": {
        "Mode": "GOVERNANCE",
        "RetainUntilDate": "2025-01-01T00:00:00.000Z"
    }
}
```

- Per i dettagli sull'API, vedere GetObjectRetentionin AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, GetObjectRetentionconsulta AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
        }
    }
```

- Per i dettagli sull'API, GetObjectRetentionconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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");
}
```

- Per i dettagli sull'API, GetObjectRetentionconsulta AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: Il comando restituisce la modalità e la data fino a quando l'oggetto non viene mantenuto.

```
Get-S3ObjectRetention -BucketName 's3buckettesting' -Key 'testfile.txt'
```

- Per i dettagli sull'API, vedere GetObjectRetentionin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetObjectTagging`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetObjectTagging`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base sui tag

CLI

AWS CLI

Per recuperare i tag allegati a un oggetto

L'`get-object-tagging`esempio seguente recupera i valori per la chiave specificata dall'oggetto specificato.

```
aws s3api get-object-tagging \
    --bucket my-bucket \
    --key doc1.rtf
```

Output:

```
{
    "TagSet": [
        {
            "Value": "confidential",
            "Key": "designation"
        }
    ]
}
```

L'`get-object-tagging`esempio seguente tenta di recuperare i set di tag dell'oggetto`doc2.rtf`, che non ha tag.

```
aws s3api get-object-tagging \
    --bucket my-bucket \
    --key doc2.rtf
```

Output:

```
{
    "TagSet": []
}
```

L'`get-object-tagging`esempio seguente recupera i set di tag dell'oggetto`doc3.rtf`, che ha più tag.

```
aws s3api get-object-tagging \
```

```
    --bucket my-bucket \
    --key doc3.rtf
```

Output:

```
{
    "TagSet": [
        {
            "Value": "confidential",
            "Key": "designation"
        },
        {
            "Value": "finance",
            "Key": "department"
        },
        {
            "Value": "payroll",
            "Key": "team"
        }
    ]
}
```

- Per i dettagli sull'API, consultate GetObjectTagging AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: l'esempio restituisce i tag associati all'oggetto presente nel bucket S3 specificato.

```
Get-S3ObjectTagSet -Key 'testfile.txt' -BucketName 'testbucket123'
```

Output:

```
Key  Value
---  -----
test value
```

- Per i dettagli sull'API, vedere GetObjectTaggingin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetPublicAccessBlock`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetPublicAccessBlock`.

CLI

AWS CLI

Per impostare o modificare la configurazione di accesso pubblico a blocchi per un bucket

L'`get-public-access-block`esempio seguente visualizza la configurazione dell'accesso pubblico a blocchi per il bucket specificato.

```
aws s3api get-public-access-block \
    --bucket my-bucket
```

Output:

```
{
    "PublicAccessBlockConfiguration": {
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "BlockPublicAcls": true,
        "RestrictPublicBuckets": true
    }
}
```

- Per i dettagli sull'API, vedere GetPublicAccessBlockin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: il comando restituisce la configurazione del blocco di accesso pubblico del bucket S3 specificato.

```
Get-S3PublicAccessBlock -BucketName 's3testbucket'
```

- Per i dettagli sull'API, vedere GetPublicAccessBlockin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **HeadBucket** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzareHeadBucket.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta HeadBucket AWS CLICommand Reference.

CLI

AWS CLI

Il comando seguente verifica l'accesso a un bucket denominato: `my-bucket`

```
aws s3api head-bucket --bucket my-bucket
```

Se il bucket esiste e si ha accesso ad esso, non viene restituito alcun output. In caso contrario, verrà visualizzato un messaggio di errore. Per esempio:

```
A client error (404) occurred when calling the HeadBucket operation: Not Found
```

- Per i dettagli sull'API, consulta HeadBucket AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, HeadBucketconsulta AWS SDK for Go API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta HeadBucket AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`HeadObject`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`HeadObject`.

CLI

AWS CLI

Il comando seguente recupera i metadati per un oggetto in un bucket denominato: `my-bucket`

```
aws s3api head-object --bucket my-bucket --key index.html
```

Output:

```
{
    "AcceptRanges": "bytes",
    "ContentType": "text/html",
    "LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
    "ContentLength": 77,
    "VersionId": "null",
    "ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
    "Metadata": {}
}
```

- Per i dettagli sull'API, consultate Command HeadObjectReference AWS CLI .

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Determinazione del tipo di contenuto di un oggetto.

```
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);
        }
    }
}
```

Controllo dello stato di ripristino di un oggetto.

```
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);
        }
    }
}
```

- Per i dettagli sull'API, HeadObjectconsulta AWS SDK for Java 2.x API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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__
```

- Per i dettagli sull'API, HeadObjectconsulta AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListBucketAnalyticsConfigurations`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListBucketAnalyticsConfigurations`.

CLI

AWS CLI

Per recuperare un elenco di configurazioni di analisi per un bucket

Quanto segue `list-bucket-analytics-configurations` recupera un elenco di configurazioni di analisi per il bucket specificato.

```
aws s3api list-bucket-analytics-configurations \
    --bucket my-bucket
```

Output:

```
{
    "AnalyticsConfigurationList": [
        {
            "StorageClassAnalysis": {},
            "Id": "1"
        }
    ],
    "IsTruncated": false
}
```

- Per i dettagli sull'API, consulta Command ListBucketAnalyticsConfigurationsReference AWS CLI .

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce le prime 100 configurazioni di analisi del bucket S3 specificato.

```
Get-S3BucketAnalyticsConfigurationList -BucketName 's3casetestbucket'
```

- Per i dettagli sull'API, vedere ListBucketAnalyticsConfigurationsin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListBucketInventoryConfigurations`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListBucketInventoryConfigurations`.

CLI

AWS CLI

Per recuperare un elenco di configurazioni di inventario per un bucket

L'`list-bucket-inventory-configurations`esempio seguente elenca le configurazioni di inventario per il bucket specificato.

```
aws s3api list-bucket-inventory-configurations \
    --bucket my-bucket
```

Output:

```
{
    "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
}
```

- Per i dettagli sull'API, vedere ListBucketInventoryConfigurationsin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce le prime 100 configurazioni di inventario del bucket S3 specificato.

```
Get-S3BucketInventoryConfigurationList -BucketName 's3testbucket'
```

- Per i dettagli sull'API, vedere ListBucketInventoryConfigurationsin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListBuckets`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListBuckets`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
}
```

- Per i dettagli sull'API, ListBucketsconsulta AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
}
```

- Per i dettagli sull'API, ListBucketsconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Il comando seguente utilizza il `list-buckets` comando per visualizzare i nomi di tutti i bucket Amazon S3 (in tutte le regioni):

```
aws s3api list-buckets --query "Buckets[].Name"
```

L'opzione query filtra l'output dei `list-buckets` soli nomi dei bucket.

Per ulteriori informazioni sui bucket, consulta Working with Amazon S3 Buckets nella Amazon S3 Developer Guide.

- Per i dettagli sull'API, consulta ListBucketsCommand Reference.AWS CLI

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, ListBucketsconsulta AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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());
        }
    }
}
```

- Per i dettagli sull'API, ListBucketsconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca i bucket.

```
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);
```

```
  }
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, ListBucketsconsulta AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando restituisce tutti i bucket S3.

```
Get-S3Bucket
```

Esempio 2: questo comando restituisce un bucket denominato «test-files»

```
Get-S3Bucket -BucketName test-files
```

- Per i dettagli sull'API, vedere ListBucketsin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta ListBuckets AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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__
```

- Per i dettagli sull'API, ListBucketsconsulta AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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(())
}
```

- Per i dettagli sulle API, consulta la ListBucketsguida di riferimento all'API AWS SDK for Rust.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta la ListBucketsguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListMultipartUploads`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListMultipartUploads`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Elimina caricamenti multiparte incompleti

CLI

AWS CLI

Il comando seguente elenca tutti i caricamenti multiparte attivi per un bucket denominato: `my-bucket`

```
aws s3api list-multipart-uploads --bucket my-bucket
```

Output:

```
{
    "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": []
}
```

In corso, i caricamenti multiparte comportano costi di archiviazione in Amazon S3. Completa o annulla un caricamento multiparte attivo per rimuoverne le parti dal tuo account.

- Per i dettagli sull'API, consulta AWS CLI Command ListMultipartUploadsReference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
```

```
    }
}
```

- Per i dettagli sull'API, ListMultipartUploadsconsulta AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListObjectVersions`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListObjectVersions`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Utilizzo degli oggetti con versione

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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}'");
            }
        }
    }
```

- Per i dettagli sull'API, ListObjectVersionsconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Il comando seguente recupera le informazioni sulla versione di un oggetto in un bucket denominato: `my-bucket`

```
aws s3api list-object-versions --bucket my-bucket --prefix index.html
```

Output:


```
{
    "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
        }
    ]
}
```

- Per i dettagli sull'API, consultate AWS CLI Command ListObjectVersionsReference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, ListObjectVersionsconsulta AWS SDK for Go API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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(())
}
```

- Per i dettagli sulle API, consulta la ListObjectVersionsguida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListObjects`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListObjects`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creare una pagina Web che elenca gli oggetti Amazon S3

CLI

AWS CLI

L'esempio seguente utilizza il `list-objects` comando per visualizzare i nomi di tutti gli oggetti nel bucket specificato:

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size:
 Size}'
```

L'esempio utilizza l'`--query`argomento per filtrare l'output di `list-objects` fino al valore e alla dimensione della chiave per ogni oggetto

Per ulteriori informazioni sugli oggetti, consulta Working with Amazon S3 Objects nella Amazon S3 Developer Guide.

- Per i dettagli sull'API, consulta AWS CLI Command ListObjectsReference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando recupera le informazioni su tutti gli elementi nel bucket «test-files».

```
Get-S3Object -BucketName test-files
```

Esempio 2: questo comando recupera le informazioni sull'elemento "sample.txt" dal bucket «test-files».

```
Get-S3Object -BucketName test-files -Key sample.txt
```

Esempio 3: Questo comando recupera le informazioni su tutti gli elementi con il prefisso «sample» dal bucket «test-files».

```
Get-S3Object -BucketName test-files -KeyPrefix sample
```

- Per i dettagli sull'API, vedere in Cmdlet Reference. ListObjectsAWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListObjectsV2`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListObjectsV2`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base su bucket e oggetti

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
            }
        }
```

Elencare gli oggetti con un impaginatore.

```
    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}");
                }
            }
        }
    }
```

- Per i dettagli sull'API, consulta la ListObjectsversione V2 in AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, vedi ListObjectsV2 in AWS CLI Command Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
bool AwsDoc::S3::listObjects(const Aws::String &bucketName,
                             const Aws::S3::S3ClientConfiguration &clientConfig)
 {
    Aws::S3::S3Client s3Client(clientConfig);

    Aws::S3::Model::ListObjectsV2Request request;
    request.WithBucket(bucketName);

    Aws::String continuationToken; // Used for pagination.
    Aws::Vector<Aws::S3::Model::Object> allObjects;

    do {
```

```
        if (!continuationToken.empty()) {
            request.SetContinuationToken(continuationToken);
        }

        auto outcome = s3Client.ListObjectsV2(request);

        if (!outcome.IsSuccess()) {
            std::cerr << "Error: listObjects: " <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        } else {
            Aws::Vector<Aws::S3::Model::Object> objects =
                    outcome.GetResult().GetContents();

            allObjects.insert(allObjects.end(), objects.begin(), objects.end());
            continuationToken = outcome.GetResult().GetNextContinuationToken();
        }
    } while (!continuationToken.empty());

    std::cout << allObjects.size() << " object(s) found:" << std::endl;

    for (const auto &object: allObjects) {
        std::cout << "  " << object.GetKey() << std::endl;
    }

    return true;
}
```

- Per i dettagli sull'API, consulta la ListObjectsversione V2 in AWS SDK for C++ API Reference.

CLI

AWS CLI

Per ottenere un elenco di oggetti in un bucket

L'`list-objects-v2`esempio seguente elenca gli oggetti nel bucket specificato.

```
aws s3api list-objects-v2 \
    --bucket my-bucket
```

Output:

```
{
    "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
        }
    ]
}
```

- Per i dettagli sull'API, vedere ListObjectsV2 in AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta la ListObjectsversione V2 in AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
    }
}
```

Elenca gli oggetti usando l'impaginazione.

```
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);
        }
    }
}
```

- Per i dettagli sull'API, consulta la ListObjectsversione V2 in AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca tutti gli oggetti nel bucket. Se è presente più di un oggetto, IsTruncated NextContinuationToken verrà utilizzato per scorrere l'elenco completo.

```
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);
  }
};
```

- Per i dettagli sull'API, consulta la versione ListObjectsV2 in AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta ListObjectsV2 in AWS SDK per il riferimento all'API Kotlin.

PHP

SDK per PHP

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca gli oggetti in 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.");
        }
```

- Per i dettagli sull'API, consulta la ListObjectsversione V2 in AWS SDK for PHP API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta ListObjectsV2 nella Guida di riferimento all'API AWS SDK for Python (Boto3).

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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__
```

- Per i dettagli sull'API, consulta la ListObjectsversione V2 in AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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(())
}
```

- Per i dettagli sull'API, consulta la ListObjectsversione 2 in AWS SDK for Rust API reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.
```

- Per i dettagli sull'API, consulta ListObjectsV2 in AWS SDK per il riferimento all'API SAP ABAP.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
    }
```

- Per i dettagli sull'API, consulta la ListObjectsversione 2 in AWS SDK for Swift API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutBucketAccelerateConfiguration`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutBucketAccelerateConfiguration`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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");
            }
        }
    }
```

- Per i dettagli sull'API, PutBucketAccelerateConfigurationconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Per impostare la configurazione di accelerazione di un bucket

L'`put-bucket-accelerate-configuration`esempio seguente abilita la configurazione di accelerazione per il bucket specificato.

```
aws s3api put-bucket-accelerate-configuration \
    --bucket my-bucket \
    --accelerate-configuration Status=Enabled
```

Questo comando non produce alcun output.

- Per i dettagli sull'API, vedere PutBucketAccelerateConfigurationin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando abilita l'accelerazione di trasferimento per il bucket S3 specificato.

```
$statusVal = New-Object Amazon.S3.BucketAccelerateStatus('Enabled')
Write-S3BucketAccelerateConfiguration -BucketName 's3testbucket' -
AccelerateConfiguration_Status $statusVal
```

- Per i dettagli sull'API, vedere PutBucketAccelerateConfigurationin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutBucketAcl`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutBucketAcl`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestire le liste di controllo degli accessi (ACL)

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
        }
```

- Per i dettagli sull'API, PutBucketAclconsulta AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
}
```

- Per i dettagli sull'API, PutBucketAclconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Questo esempio concede `full control` a due AWS utenti (user1@example.com e user2@example.com) `read` l'autorizzazione a tutti:

```
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
```

Vedi http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html per i dettagli sugli ACL personalizzati (i comandi ACL di s3api, ad esempio`put-bucket-acl`, usano la stessa notazione abbreviata degli argomenti).

- Per i dettagli sull'API, consulta Command Reference. PutBucketAclAWS CLI

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
}
```

- Per i dettagli sull'API, PutBucketAclconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Inserisci l'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);
  }
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, PutBucketAclconsulta AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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")
    }
}
```

- Per i dettagli sull'API, PutBucketAclconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta PutBucketAcl AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutBucketCors`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutBucketCors`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
```

- Per i dettagli sull'API, PutBucketCorsconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

L'esempio seguente abilita PUT e DELETE richiede da www.example.com e abilita GET le richieste da qualsiasi dominio: POST

```
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
    }
  ]
}
```

- Per i dettagli sull'API, consulta Command PutBucketCorsReference AWS CLI .

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
}
```

- Per i dettagli sull'API, PutBucketCorsconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Aggiungi una regola 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);
  }
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, PutBucketCorsconsulta AWS SDK for JavaScript API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta PutBucketCors AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, PutBucketCorsconsulta AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutBucketEncryption`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutBucketEncryption`.

CLI

AWS CLI

Per configurare la crittografia lato server per un bucket

L'`put-bucket-encryption`esempio seguente imposta la crittografia AES256 come predefinita per il bucket specificato.

```
aws s3api put-bucket-encryption \
    --bucket my-bucket \
    --server-side-encryption-configuration '{"Rules":
 [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}}]}'
```

Questo comando non produce alcun output.

- Per i dettagli sull'API, vedere PutBucketEncryptionin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando abilita la crittografia AES256 lato server predefinita con Amazon S3 Managed Keys (SSE-S3) sul bucket specificato.

```
$Encryptionconfig = @{ServerSideEncryptionByDefault =
 @{ServerSideEncryptionAlgorithm = "AES256"}}
Set-S3BucketEncryption -BucketName 's3testbucket' -
ServerSideEncryptionConfiguration_ServerSideEncryptionRule $Encryptionconfig
```

- Per i dettagli sull'API, vedere in Cmdlet Reference. PutBucketEncryptionAWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutBucketLifecycleConfiguration`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutBucketLifecycleConfiguration`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Elimina caricamenti multiparte incompleti

- Utilizzo degli oggetti con versione

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
```

- Per i dettagli sull'API, PutBucketLifecycleConfigurationconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Il comando seguente applica una configurazione del ciclo di vita a un bucket denominato: `my-bucket`

```
aws s3api put-bucket-lifecycle-configuration --bucket my-bucket --lifecycle-
configuration  file://lifecycle.json
```

Il file `lifecycle.json` è un documento JSON nella cartella corrente che specifica due regole:

```
{
    "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 prima regola sposta i file con il prefisso `rotated` su Glacier nella data specificata. La seconda regola sposta le vecchie versioni degli oggetti su Glacier quando non sono più attuali.

Per informazioni sui formati di timestamp accettabili, consulta Specificare i valori dei parametri nella Guida per l'utente della CLI AWS .

- Per i dettagli sull'API, consulta PutBucketLifecycleConfigurationCommand Reference.AWS CLI

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
                }
        }
}
```

- Per i dettagli sull'API, PutBucketLifecycleConfigurationconsulta AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta PutBucketLifecycleConfiguration AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **PutBucketLogging** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutBucketLogging`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
        }
    }
```

- Per i dettagli sull'API, PutBucketLoggingconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Esempio 1: per impostare la registrazione delle policy sui bucket

L'`put-bucket-logging`esempio seguente imposta la politica di registrazione per. MyBucket Innanzitutto, concedi al servizio di registrazione l'autorizzazione principale nella tua policy del bucket utilizzando il comando. `put-bucket-policy`

```
aws s3api put-bucket-policy \
    --bucket MyBucket \
    --policy file://policy.json
```

Contenuto di `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"}
            }
        }
    ]
}
```

Per applicare la politica di registrazione, usa. `put-bucket-logging`

```
aws s3api put-bucket-logging \
    --bucket MyBucket \
    --bucket-logging-status file://logging.json
```

Contenuto di `logging.json`.

```
{
     "LoggingEnabled": {
         "TargetBucket": "MyBucket",
         "TargetPrefix": "Logs/"
     }
 }
```

Il `put-bucket-policy` comando è necessario per concedere `s3:PutObject` le autorizzazioni al responsabile del servizio di registrazione.

Per ulteriori informazioni, consulta Amazon S3 Server Access Logging nella Amazon S3 User Guide.

Esempio 2: impostare una policy sui bucket per registrare l'accesso a un solo utente

L'`put-bucket-logging`esempio seguente imposta la politica di registrazione per. MyBucket L' AWS utente bob@example.com avrà il pieno controllo sui file di registro e nessun altro potrà accedervi. Innanzitutto, concedi l'autorizzazione a S3 con`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
```

Quindi applica la politica di registrazione utilizzando. `put-bucket-logging`

```
aws s3api put-bucket-logging \
    --bucket MyBucket \
    --bucket-logging-status file://logging.json
```

Contenuto di `logging.json`.

```
{
```

```
    "LoggingEnabled": {
        "TargetBucket": "MyBucket",
        "TargetPrefix": "MyBucketLogs/",
        "TargetGrants": [
            {
                "Grantee": {
                    "Type": "AmazonCustomerByEmail",
                    "EmailAddress": "bob@example.com"
                },
                "Permission": "FULL_CONTROL"
            }
        ]
    }
}
```

il `put-bucket-acl` comando è necessario per concedere al sistema di consegna dei log di S3 le autorizzazioni necessarie (autorizzazioni di scrittura e lettura acp).

Per ulteriori informazioni, consulta Amazon S3 Server Access Logging nella Amazon S3 Developer Guide.

- Per i dettagli sull'API, consulta Command PutBucketLoggingReference AWS CLI .

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutBucketNotification`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutBucketNotification`.

CLI

AWS CLI

Applica una configurazione di notifica a un bucket denominato: `my-bucket`

```
aws s3api put-bucket-notification --bucket my-bucket --notification-configuration
 file://notification.json
```

Il file `notification.json` è un documento JSON nella cartella corrente che specifica un argomento SNS e un tipo di evento da monitorare:

```
{
  "TopicConfiguration": {
    "Event": "s3:ObjectCreated:*",
    "Topic": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic"
  }
}
```

L'argomento SNS deve avere una policy IAM allegata che consenta ad Amazon S3 di pubblicare su di esso:

```
{
 "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"
    }
   }
  }
 ]
}
```

- Per i dettagli sull'API, consulta AWS CLI Command PutBucketNotificationReference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio configura la configurazione dell'argomento SNS per l'evento S3 ObjectRemovedDelete e abilita la notifica per il bucket s3 specificato

```
$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
```

Esempio 2: questo esempio abilita le notifiche ObjectCreatedAll relative al bucket specificato che lo invia alla funzione 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
```

Esempio 3: Questo esempio crea 2 diverse configurazioni Lambda sulla base di suffissi chiave diversi e configurate entrambe in un unico 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
```

- Per i dettagli sull'API, vedere PutBucketNotificationin Cmdlet Reference.AWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **PutBucketNotificationConfiguration** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzarePutBucketNotificationConfiguration.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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}");
            }
        }
    }
```

- Per i dettagli sull'API, PutBucketNotificationConfigurationconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Per abilitare le notifiche specificate in un bucket

L'`put-bucket-notification-configuration`esempio seguente applica una configurazione di notifica a un bucket denominato. `my-bucket` Il file `notification.json` è un documento JSON nella cartella corrente che specifica un argomento SNS e un tipo di evento da monitorare.

```
aws s3api put-bucket-notification-configuration \
    --bucket my-bucket \
    --notification-configuration file://notification.json
```

Contenuto di `notification.json`.

```
{
    "TopicConfigurations": [
        {
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:s3-notification-
topic",
            "Events": [
                "s3:ObjectCreated:*"
            ]
        }
    ]
}
```

L'argomento SNS deve avere una policy IAM allegata che consenta ad Amazon S3 di pubblicare su di esso.

```
{
    "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"
                }
            }
        }
    ]
}
```

- Per i dettagli sull'API, consulta AWS CLI Command PutBucketNotificationConfigurationReference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.Event;
import software.amazon.awssdk.services.s3.model.NotificationConfiguration;
```

```
import
 software.amazon.awssdk.services.s3.model.PutBucketNotificationConfigurationRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.TopicConfiguration;
import java.util.ArrayList;
import java.util.List;

public class SetBucketEventBridgeNotification {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <bucketName>\s

                Where:
                    bucketName - The Amazon S3 bucket.\s
                    topicArn - The Simple Notification Service topic ARN.\s
                    id - An id value used for the topic configuration. This value
 is displayed in the AWS Management Console.\s
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String topicArn = args[1];
        String id = args[2];
        Region region = Region.US_EAST_1;
        S3Client s3Client = S3Client.builder()
                .region(region)
                .build();

        setBucketNotification(s3Client, bucketName, topicArn, id);
        s3Client.close();
    }

    public static void setBucketNotification(S3Client s3Client, String
 bucketName, String topicArn, String id) {
        try {
            List<Event> events = new ArrayList<>();
            events.add(Event.S3_OBJECT_CREATED_PUT);
```

```
            TopicConfiguration config = TopicConfiguration.builder()
                    .topicArn(topicArn)
                    .events(events)
                    .id(id)
                    .build();

            List<TopicConfiguration> topics = new ArrayList<>();
            topics.add(config);

            NotificationConfiguration configuration =
 NotificationConfiguration.builder()
                    .topicConfigurations(topics)
                    .build();

            PutBucketNotificationConfigurationRequest configurationRequest =
 PutBucketNotificationConfigurationRequest
                    .builder()
                    .bucket(bucketName)
                    .notificationConfiguration(configuration)
                    .skipDestinationValidation(true)
                    .build();

            // Set the bucket notification configuration.
            s3Client.putBucketNotificationConfiguration(configurationRequest);
            System.out.println("Added bucket " + bucketName + " with EventBridge
 events enabled.");

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, PutBucketNotificationConfigurationconsulta AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **PutBucketPolicy** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutBucketPolicy`.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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"
            "}";
}
```

- Per i dettagli sull'API, PutBucketPolicyconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Questo esempio consente a tutti gli utenti di recuperare qualsiasi oggetto in MyBucket. MySecretFolder Inoltre concede `put` l'`delete`autorizzazione all'utente root dell' AWS account: `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/*"
      }
   ]
}
```

- Per i dettagli sull'API, consulta PutBucketPolicy AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
    }
}
```

- Per i dettagli sull'API, PutBucketPolicyconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Aggiungi la policy.

```
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);
  }
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, PutBucketPolicyconsulta AWS SDK for JavaScript API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta PutBucketPolicy AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, PutBucketPolicyconsulta AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutBucketReplication`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutBucketReplication`.

CLI

AWS CLI

Per configurare la replica per un bucket S3

L'`put-bucket-replication`esempio seguente applica una configurazione di replica al bucket S3 specificato.

```
aws s3api put-bucket-replication \
```

```
    --bucket AWSDOC-EXAMPLE-BUCKET1 \
    --replication-configuration file://replication.json
```

Contenuto di `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"
            }
        }
    ]
}
```

Il bucket di destinazione deve avere il controllo delle versioni abilitato. Il ruolo specificato deve avere l'autorizzazione a scrivere nel bucket di destinazione e avere una relazione di trust che consenta ad Amazon S3 di assumere il ruolo.

Esempio di politica di autorizzazione dei ruoli:

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

Esempio di politica sulle relazioni di fiducia:

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

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta il titolo dell'argomento nella Guida per l'utente della console di Amazon Simple Storage Service.

- Per i dettagli sull'API, consulta PutBucketReplication AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo esempio imposta una configurazione di replica con un'unica regola che consente la replica nel bucket 'exampletargetbucket' di tutti i nuovi oggetti creati con il prefisso del nome chiave "" nel bucket 'examplebucket'. TaxDocs

```
$rule1 = New-Object Amazon.S3.Model.ReplicationRule
$rule1.ID = "Rule-1"
$rule1.Status = "Enabled"
$rule1.Prefix = "TaxDocs"
$rule1.Destination = @{ BucketArn = "arn:aws:s3:::exampletargetbucket" }

$params = @{
    BucketName = "examplebucket"
    Configuration_Role = "arn:aws:iam::35667example:role/
CrossRegionReplicationRoleForS3"
    Configuration_Rule = $rule1
}

Write-S3BucketReplication @params
```

Esempio 2: questo esempio imposta una configurazione di replica con più regole che consentono la replica nel bucket 'exampletargetbucket' di qualsiasi nuovo oggetto creato con il prefisso del nome chiave "TaxDocs" o OtherDocs "». I prefissi chiave non devono sovrapporsi.

```
$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
```

Esempio 3: questo esempio aggiorna la configurazione di replica nel bucket specificato per disabilitare la regola che controlla la replica degli oggetti con il prefisso del nome chiave "" nel bucket 'exampletargetbucket'. TaxDocs

```
$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
```

- Per PutBucketReplication AWS Tools for PowerShelli dettagli sull'API, vedere in Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutBucketRequestPayment`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutBucketRequestPayment`.

CLI

AWS CLI

Esempio 1: abilitare la configurazione ``requester pays`` per un bucket

L'esempio seguente abilita il bucket specificatoput-bucket-request-payment. `requester pays`

```
aws s3api put-bucket-request-payment \
    --bucket my-bucket \
    --request-payment-configuration '{"Payer":"Requester"}'
```

Questo comando non produce alcun output.

Esempio 2: disabilitare la configurazione ``requester pays`` per un bucket

L'esempio seguente disabilita per il bucket specificatoput-bucket-request-payment. `requester pays`

```
aws s3api put-bucket-request-payment \
    --bucket my-bucket \
    --request-payment-configuration '{"Payer":"BucketOwner"}'
```

Questo comando non produce alcun output.

- Per i dettagli sull'API, vedere PutBucketRequestPaymentin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: aggiorna la configurazione del pagamento della richiesta per il bucket denominato «mybucket» in modo che alla persona che richiede i download dal bucket venga addebitato il costo del download. Per impostazione predefinita, il proprietario del bucket paga per i download. Per riportare il pagamento della richiesta ai valori predefiniti, usa 'BucketOwner' per il parametro RequestPaymentConfiguration _Payer.

```
Write-S3BucketRequestPayment -BucketName mybucket -
RequestPaymentConfiguration_Payer Requester
```

- Per i dettagli sull'API, vedere PutBucketRequestPaymentin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutBucketTagging`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutBucketTagging`.

CLI

AWS CLI

Il comando seguente applica una configurazione di tag a un bucket denominato: `my-bucket`

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging file://tagging.json
```

Il file `tagging.json` è un documento JSON nella cartella corrente che specifica i tag:

```
{
   "TagSet": [
     {
       "Key": "organization",
       "Value": "marketing"
     }
   ]
}
```

Oppure applica una configurazione di tagging `my-bucket` direttamente dalla riga di comando:

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging
 'TagSet=[{Key=organization,Value=marketing}]'
```

- Per i dettagli sull'API, consulta PutBucketTagging AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando applica due tag a un bucket denominato**`cloudtrail-test-2018`**: un tag con una chiave di Stage e un valore di Test e un tag con una chiave di Environment e un valore di Alpha. Per verificare che i tag siano stati aggiunti al bucket, esegui. **`Get-S3BucketTagging -BucketName bucket_name`** I risultati dovrebbero mostrare i tag che hai applicato al bucket nel primo comando. Nota che **`Write-S3BucketTagging`** sovrascrive l'intero set di tag esistente su un bucket. Per aggiungere o eliminare singoli tag, eseguire i cmdlet Resource Groups and Tagging API e. **`Add-RGTResourceTag`** **`Remove-RGTResourceTag`** In alternativa, utilizza Tag Editor nella Console di AWS gestione per gestire i tag bucket S3.

```
Write-S3BucketTagging -BucketName cloudtrail-test-2018 -TagSet @( @{ Key="Stage";
 Value="Test" }, @{ Key="Environment"; Value="Alpha" } )
```

Esempio 2: questo comando reindirizza un bucket denominato **`cloudtrail-test-2018`** nel cmdlet. **`Write-S3BucketTagging`** Applica i tag Stage:Production e Department:Finance al bucket. Nota che **`Write-S3BucketTagging`** sovrascrive l'intero set di tag esistente su un bucket.

```
Get-S3Bucket -BucketName cloudtrail-test-2018 | Write-S3BucketTagging
 -TagSet @( @{ Key="Stage"; Value="Production" }, @{ Key="Department";
 Value="Finance" } )
```

- Per i dettagli sull'API, vedere PutBucketTaggingin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutBucketVersioning`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutBucketVersioning`.

CLI

AWS CLI

Il comando seguente abilita il controllo delle versioni su un bucket denominato: `my-bucket`

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-configuration
 Status=Enabled
```

Il comando seguente abilita il controllo delle versioni e utilizza un codice mfa

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-configuration
 Status=Enabled --mfa "SERIAL 123456"
```

- Per i dettagli sull'API, vedere PutBucketVersioningin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: il comando abilita il controllo delle versioni per il bucket S3 specificato.

```
Write-S3BucketVersioning -BucketName 's3testbucket' -VersioningConfig_Status
 Enabled
```

- Per i dettagli sull'API, vedere PutBucketVersioningin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutBucketWebsite`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutBucketWebsite`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
                // Put the website configuration.
                PutBucketWebsiteRequest putRequest = new
PutBucketWebsiteRequest()
                {
                    BucketName = bucketName,
                    WebsiteConfiguration = new WebsiteConfiguration()
                    {
                        IndexDocumentSuffix = indexDocumentSuffix,
                        ErrorDocument = errorDocument,
                    },
                };
                PutBucketWebsiteResponse response = await
client.PutBucketWebsiteAsync(putRequest);
```

- Per i dettagli sull'API, PutBucketWebsiteconsulta AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
}
```

- Per i dettagli sull'API, PutBucketWebsiteconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Applica una configurazione statica del sito Web a un bucket denominato`my-bucket`:

```
aws s3api put-bucket-website --bucket my-bucket --website-configuration file://
website.json
```

Il file `website.json` è un documento JSON nella cartella corrente che specifica l'indice e le pagine di errore per il sito Web:

```
{
    "IndexDocument": {
        "Suffix": "index.html"
    },
    "ErrorDocument": {
        "Key": "error.html"
    }
}
```

- Per i dettagli sull'API, consulta AWS CLI Command PutBucketWebsiteReference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
}
```

- Per i dettagli sull'API, PutBucketWebsiteconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Imposta la configurazione del sito 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);
  }
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, PutBucketWebsiteconsulta AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: il comando abilita l'hosting del sito Web per il bucket specificato con il documento indice come 'index.html' e il documento di errore come 'error.html'.

```
Write-S3BucketWebsite -BucketName 's3testbucket' -
WebsiteConfiguration_IndexDocumentSuffix 'index.html' -
WebsiteConfiguration_ErrorDocument 'error.html'
```

- Per i dettagli sull'API, vedere in Cmdlet Reference. PutBucketWebsiteAWS Tools for PowerShell

Ruby

SDK per Ruby

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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__
```

- Per i dettagli sull'API, PutBucketWebsiteconsulta AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutObject`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutObject`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Nozioni di base su bucket e oggetti
- Tieni traccia dei caricamenti e dei download

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
            }
        }
```

Caricare un oggetto con la crittografia lato server.

```
    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");
            }
        }
    }
```

- Per i dettagli sull'API, PutObjectconsulta AWS SDK for .NET API Reference.

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta PutObject AWS CLICommand Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
}
```

- Per i dettagli sull'API, PutObjectconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

L'esempio seguente utilizza il `put-object` comando per caricare un oggetto su Amazon S3:

```
aws s3api put-object --bucket text-content --key dir-1/my_images.tar.bz2 --body
 my_images.tar.bz2
```

L'esempio seguente mostra il caricamento di un file video (il file video viene specificato utilizzando la sintassi del file system 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
```

Per ulteriori informazioni sul caricamento di oggetti, consulta Uploading Objects nella Amazon S3 Developer Guide.

- Per i dettagli sull'API, consulta Command PutObjectReference AWS CLI .

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Metti un oggetto in un bucket utilizzando l'API di basso livello.

```
// 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
}
```

Carica un oggetto in un bucket utilizzando un gestore di trasferimenti.

```
// 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
}
```

- Per i dettagli sull'API, consulta la sezione PutObject AWS SDK for GoAPI Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Carica un file in un bucket utilizzando un'interfaccia 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);
        }
    }
}
```

Usa un S3 TransferManager per caricare un file in un bucket. Visualizza il file completo ed esegui il test.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedFileUpload;
import software.amazon.awssdk.transfer.s3.model.FileUpload;
import software.amazon.awssdk.transfer.s3.model.UploadFileRequest;
import software.amazon.awssdk.transfer.s3.progress.LoggingTransferListener;
import java.net.URI;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.file.Paths;
import java.util.UUID;

    public String uploadFile(S3TransferManager transferManager, String
 bucketName,
                             String key, URI filePathURI) {
        UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
            .putObjectRequest(b -> b.bucket(bucketName).key(key))
            .source(Paths.get(filePathURI))
            .build();

        FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);
```

```
        CompletedFileUpload uploadResult = fileUpload.completionFuture().join();
        return uploadResult.response().eTag();
    }
```

Caricare un oggetto in un bucket e impostare tag mediante un'interfaccia 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;
    }
}
```

Caricare un oggetto in un bucket e impostare metadati mediante un'interfaccia 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);
        }
    }
}
```

Caricare un oggetto in un bucket e impostare un valore di conservazione per l'oggetto mediante un'interfaccia 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);
        }
    }
}
```

- Per i dettagli sull'API, consulta la sezione AWS SDK for Java 2.x API PutObjectReference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Carica l'oggetto.

```
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);
  }
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, PutObjectconsulta AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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}")
    }
}
```

- Per i dettagli sull'API, PutObjectconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Carica un oggetto in 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.");
        }
```

- Per i dettagli sull'API, PutObjectconsulta AWS SDK for PHP API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: questo comando carica il singolo file "local-sample.txt" su Amazon S3, creando un oggetto con la chiave "sample.txt" nel bucket «test-files».

```
Write-S3Object -BucketName test-files -Key "sample.txt" -File .\local-sample.txt
```

Esempio 2: questo comando carica il singolo file "sample.txt" su Amazon S3, creando un oggetto con la chiave "sample.txt" nel bucket «test-files». Se il parametro -Key non viene fornito, il nome del file viene utilizzato come chiave dell'oggetto S3.

```
Write-S3Object -BucketName test-files -File .\sample.txt
```

Esempio 3: questo comando carica il singolo file "local-sample.txt" su Amazon S3, creando un oggetto con la chiave "prefix/to/sample.txt" nel bucket «test-files».

```
Write-S3Object -BucketName test-files -Key "prefix/to/sample.txt" -File .\local-
sample.txt
```

Esempio 4: questo comando carica tutti i file nella sottodirectory «Scripts» nel bucket «test-files» e applica il prefisso chiave comune "" a ciascun oggetto. SampleScripts Ogni file caricato avrà una chiave "SampleScripts/filename" dove 'filename' varia.

```
Write-S3Object -BucketName test-files -Folder .\Scripts -KeyPrefix SampleScripts\
```

Esempio 5: questo comando carica tutti i file*.ps1 nella directory locale «Scripts» nel bucket «test-files» e applica il prefisso chiave comune "" a ciascun oggetto. SampleScripts Ogni file caricato avrà una chiave "/filename.ps1" dove 'filename' varia. SampleScripts

```
Write-S3Object -BucketName test-files -Folder .\Scripts -KeyPrefix SampleScripts\
 -SearchPattern *.ps1
```

Esempio 6: questo comando crea un nuovo oggetto S3 contenente la stringa di contenuto specificata con la chiave 'sample.txt'.

```
Write-S3Object -BucketName test-files -Key "sample.txt" -Content "object
 contents"
```

Esempio 7: questo comando carica il file specificato (il nome del file viene utilizzato come chiave) e applica i tag specificati al nuovo oggetto.

```
Write-S3Object -BucketName test-files -File "sample.txt" -TagSet
 @{Key="key1";Value="value1"},@{Key="key2";Value="value2"}
```

Esempio 8: questo comando carica in modo ricorsivo la cartella specificata e applica i tag specificati a tutti i nuovi oggetti.

```
Write-S3Object -BucketName test-files -Folder . -KeyPrefix "TaggedFiles" -Recurse
 -TagSet @{Key="key1";Value="value1"},@{Key="key2";Value="value2"}
```

- Per i dettagli sull'API, vedere PutObjectin AWS Tools for PowerShell Cmdlet Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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()
```

- Per i dettagli sull'API, consulta PutObject AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Carica un file utilizzando un caricamento gestito (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__
```

Carica un file utilizzando 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__
```

Carica un file utilizzando Object.put e aggiungi la crittografia lato server.

```
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__
```

- Per i dettagli sull'API, PutObjectconsulta AWS SDK for Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sulle API, consulta la PutObjectguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.
```

- Per i dettagli sulle API, PutObjectconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Caricamento di un file dall'archiviazione locale in 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.from(data: fileData)

        let input = PutObjectInput(
            body: dataStream,
            bucket: bucket,
            key: key
        )
        _ = try await client.putObject(input: input)
    }
```

Caricamento del contenuto di un oggetto Swift Data in un bucket.

```
   public func createFile(bucket: String, key: String, withData data: Data)
async throws {
        let dataStream = ByteStream.from(data: data)

        let input = PutObjectInput(
            body: dataStream,
            bucket: bucket,
            key: key
        )
        _ = try await client.putObject(input: input)
    }
```

- Per i dettagli sull'API, consulta la PutObjectguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutObjectAcl`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutObjectAcl`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. Puoi vedere questa azione nel contesto nel seguente esempio di codice:

- Gestire le liste di controllo degli accessi (ACL)

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
}
```

- Per i dettagli sull'API, PutObjectAclconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Il comando seguente concede `full control` a due AWS utenti (user1@example.com e user2@example.com) `read` l'autorizzazione a tutti:

```
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
```

Vedi http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html per i dettagli sugli ACL personalizzati (i comandi ACL di s3api, ad esempio`put-object-acl`, usano la stessa notazione abbreviata degli argomenti).

- Per i dettagli sull'API, consulta Command Reference. PutObjectAclAWS CLI

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta PutObjectAcl AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutObjectLegalHold`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutObjectLegalHold`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Blocca oggetti Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
        }
    }
```

- Per i dettagli sull'API, PutObjectLegalHoldconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Per applicare una conservazione a fini legali a un oggetto

L'`put-object-legal-hold`esempio seguente imposta un Legal Hold sull'oggetto`doc1.rtf`.

```
aws s3api put-object-legal-hold \
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf \
    --legal-hold Status=ON
```

Questo comando non produce alcun output.

- Per i dettagli sull'API, consultate PutObjectLegalHold AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, PutObjectLegalHold consulta AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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 +".");
    }
```

- Per i dettagli sull'API, PutObjectLegalHoldconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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");
}
```

- Per i dettagli sull'API, PutObjectLegalHold consulta AWS SDK for JavaScript API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutObjectLockConfiguration`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutObjectLockConfiguration`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Blocca oggetti Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Imposta la configurazione di blocco degli oggetti di 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;
        }
    }
```

Imposta il periodo di conservazione predefinito di 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;
        }
    }
```

- Per i dettagli sull'API, consulta la sezione [PutObjectLockConfiguration AWS SDK for .NET](#)API Reference.

CLI

AWS CLI

Per impostare una configurazione di blocco degli oggetti su un bucket

L'`put-object-lock-configuration`esempio seguente imposta un blocco degli oggetti di 50 giorni sul bucket specificato.

```
aws s3api put-object-lock-configuration \
    --bucket my-bucket-with-object-lock \
    --object-lock-configuration '{ "ObjectLockEnabled": "Enabled", "Rule":
 { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

Questo comando non produce alcun output.

- Per i dettagli sull'API, vedere PutObjectLockConfigurationin AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Imposta la configurazione di blocco degli oggetti di 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
}
```

Imposta il periodo di conservazione predefinito di 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
}
```

- Per i dettagli sull'API, consulta la sezione PutObjectLockConfiguration AWS SDK for GoAPI Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Imposta la configurazione di blocco degli oggetti di 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()
+ "'");
```

```
        }
    }
```

Imposta il periodo di conservazione predefinito di 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
+".");
   }
```

- Per i dettagli sull'API, consulta la sezione PutObjectLockConfiguration AWS SDK for Java 2.xAPI Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Imposta la configurazione di blocco degli oggetti di 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");
}
```

Imposta il periodo di conservazione predefinito di 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");
}
```

- Per i dettagli sull'API, consulta la sezione PutObjectLockConfiguration AWS SDK for JavaScriptAPI Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutObjectRetention`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutObjectRetention`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Blocca oggetti Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
        }
    }
```

- Per i dettagli sull'API, PutObjectRetentionconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Per impostare una configurazione di conservazione degli oggetti per un oggetto

L'`put-object-retention`esempio seguente imposta una configurazione di conservazione degli oggetti per l'oggetto specificato fino al 01/01/2025.

```
aws s3api put-object-retention \
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf \
    --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate":
 "2025-01-01T00:00:00" }'
```

Questo comando non produce alcun output.

- Per i dettagli sull'API, vedere in Command Reference. PutObjectRetentionAWS CLI

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, PutObjectRetentionconsulta AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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 +".");
    }
```

- Per i dettagli sull'API, PutObjectRetentionconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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");
}
```

- Per i dettagli sull'API, PutObjectRetentionconsulta AWS SDK for JavaScript API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: il comando abilita la modalità di mantenimento della governance fino alla data '31 dicembre 2019 00:00:00 'per l'oggetto' testfile.txt 'nel bucket S3 specificato.

```
Write-S3ObjectRetention -BucketName 's3buckettesting' -Key 'testfile.txt' -
Retention_Mode GOVERNANCE -Retention_RetainUntilDate "2019-12-31T00:00:00"
```

- Per i dettagli sull'API, vedere in Cmdlet Reference. PutObjectRetentionAWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`RestoreObject`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`RestoreObject`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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}");
        }
    }
```

- Per i dettagli sull'API, RestoreObjectconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Per creare una richiesta di ripristino per un oggetto

L'`restore-object`esempio seguente ripristina l'oggetto Amazon S3 Glacier specificato per il `my-glacier-bucket` bucket per 10 giorni.

```
aws s3api restore-object \
    --bucket my-glacier-bucket \
    --key doc1.rtf \
    --restore-request Days=10
```

Questo comando non produce alcun output.

- Per i dettagli sull'API, consulta Command Reference. RestoreObjectAWS CLI

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
```

```
}
```

- Per i dettagli sull'API, RestoreObjectconsulta AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`SelectObjectContent`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`SelectObjectContent`.

CLI

AWS CLI

Per filtrare il contenuto di un oggetto Amazon S3 in base a un'istruzione SQL

L'`select-object-content`esempio seguente filtra l'oggetto `my-data-file.csv` con l'istruzione SQL specificata e invia l'output a un file.

```
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"
```

Questo comando non produce alcun output.

- Per i dettagli sull'API, vedere SelectObjectContentin AWS CLI Command Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

L'esempio seguente mostra una query che utilizza un oggetto JSON. L'esempio completo mostra anche l'uso di un oggetto 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;
        }
    }
```

- Per i dettagli sull'API, consulta la sezione SelectObjectContent AWS SDK for Java 2.xAPI Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UploadPart`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UploadPart`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Esegui un caricamento in più parti
- Utilizzo dei checksum

CLI

AWS CLI

Il comando seguente carica la prima parte di un caricamento in più parti avviato con il comando: `create-multipart-upload`

```
aws s3api upload-part --bucket my-bucket --key 'multipart/01' --part-number 1 --
body part01 --upload-id
 "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3
```

L'`body`opzione richiede il nome o il percorso di un file locale per il caricamento (non utilizzate il prefisso file://). La dimensione minima della parte è di 5 MB. L'ID di caricamento viene restituito da `create-multipart-upload` e può essere recuperato anche con`list-multipart-uploads`. Bucket e chiave vengono specificati quando si crea il caricamento multiparte.

Output:

```
{
    "ETag": "\"e868e0f4719e394144ef36531ee6824c\""
}
```

Salva il valore ETag di ogni parte per utilizzarlo in un secondo momento. Sono necessari per completare il caricamento in più parti.

- Per i dettagli sull'API, consulta UploadPart AWS CLICommand Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
```

- Per i dettagli sulle API, consulta la UploadPartguida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scenari per Amazon S3 che utilizzano SDK AWS

I seguenti esempi di codice mostrano come implementare scenari comuni in Amazon S3 con AWS SDK. Questi scenari illustrano come eseguire attività specifiche richiamando più funzioni in Amazon S3. Ogni scenario include un collegamento a GitHub, dove puoi trovare istruzioni su come configurare ed eseguire il codice.

Esempi

- Crea un URL predefinito per Amazon S3 utilizzando un SDK AWS
- Una pagina Web che elenca gli oggetti Amazon S3 utilizzando un SDK AWS
- Eliminare caricamenti multiparte incompleti su Amazon S3 utilizzando un SDK AWS
- Scaricare tutti gli oggetti da un bucket Amazon Simple Storage Service (Amazon S3) in una directory locale

- Ottieni un oggetto Amazon S3 da un punto di accesso multiregionale utilizzando un SDK AWS
- Ottieni un oggetto da un bucket Amazon S3 utilizzando un AWS SDK, specificando un'intestazione If-Modified-Since
- Inizia a usare bucket e oggetti Amazon S3 utilizzando un SDK AWS
- Inizia a utilizzare la crittografia per oggetti Amazon S3 utilizzando un SDK AWS
- Inizia a usare i tag per gli oggetti Amazon S3 utilizzando un SDK AWS
- Ottieni la configurazione di conservazione legale di un oggetto Amazon S3 utilizzando un SDK AWS
- Lavora con le funzionalità di blocco degli oggetti di Amazon S3 utilizzando un SDK AWS
- Gestisci gli elenchi di controllo degli accessi (ACL) per i bucket Amazon S3 utilizzando un SDK AWS
- Gestisci oggetti Amazon S3 con versioni in batch con una funzione Lambda utilizzando un SDK AWS
- Analizza gli URI di Amazon S3 utilizzando un SDK AWS
- Esegui una copia multiparte di un oggetto Amazon S3 utilizzando un SDK AWS
- Esegui un caricamento multiparte di un oggetto Amazon S3 utilizzando un SDK AWS
- Tieni traccia del caricamento o del download di un oggetto Amazon S3 utilizzando un SDK AWS
- Esempi di approcci per i test di unità e integrazione con un SDK AWS
- Caricare in modo ricorsivo una directory locale in un bucket Amazon Simple Storage Service (Amazon S3)
- Carica o scarica file di grandi dimensioni da e verso Amazon S3 utilizzando un SDK AWS
- Carica uno stream di dimensioni sconosciute su un oggetto Amazon S3 utilizzando un SDK AWS
- Usa i checksum per lavorare con un oggetto Amazon S3 utilizzando un SDK AWS
- Lavora con oggetti con versione di Amazon S3 utilizzando un SDK AWS

Crea un URL predefinito per Amazon S3 utilizzando un SDK AWS

Gli esempi di codice seguenti mostrano come creare un URL prefirmato per Amazon S3 e caricare un oggetto.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Genera un URL prefirmato in grado di eseguire un'operazione Amazon S3 per un periodo di tempo limitato.

```
    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;
        }
    }
```

Genera un URL prefirmato ed esegui un caricamento utilizzando quell'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 per C++

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Genera un URL prefirmato per scaricare un oggetto.

```
//! 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);
}
```

Scarica usando 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;
```

```
}
```

Genera un URL prefirmato per caricare un oggetto.

```
//! 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);
}
```

Carica usando 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 per Go V2

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea funzioni che eseguono il wrap delle operazioni S3 di prefirma.

```
// 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
}
```

Esegui un esempio interattivo che genera e utilizza URL prefirmati per caricare, scaricare ed eliminare un oggetto 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))
}
```

Definisci un wrapper di richieste HTTP utilizzato dall'esempio per effettuare richieste 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 per Java 2.x

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Genera un URL prefirmato per un oggetto, quindi scaricalo (richiesta GET).

Importazioni.

```
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;
```

Genera l'URL.

```
   /* Create a pre-signed URL to download an object in a subsequent GET request.
*/
   public String createPresignedGetUrl(String bucketName, String keyName) {
       try (S3Presigner presigner = S3Presigner.create()) {

           GetObjectRequest objectRequest = GetObjectRequest.builder()
                   .bucket(bucketName)
                   .key(keyName)
                   .build();

           GetObjectPresignRequest presignRequest =
GetObjectPresignRequest.builder()
                   .signatureDuration(Duration.ofMinutes(10))  // The URL will
expire in 10 minutes.
                   .getObjectRequest(objectRequest)
                   .build();
```

```
            PresignedGetObjectRequest presignedRequest =
presigner.presignGetObject(presignRequest);
            logger.info("Presigned URL: [{}]",
presignedRequest.url().toString());
            logger.info("HTTP method: [{}]",
presignedRequest.httpRequest().method());

            return presignedRequest.url().toExternalForm();
        }
    }
```

Scaricate l'oggetto utilizzando uno dei tre approcci seguenti.

Usa la classe JDK `HttpURLConnection` (dalla v1.1) per eseguire il download.

```
    /* 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();
    }
```

Usa la classe JDK `HttpClient` (dalla v11) per eseguire il download.

```
    /* 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();
    }
```

Utilizzate la classe AWS SDK for `SdkHttpClient` Java per eseguire il download.

```
    /* 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();
    }
```

Genera un URL prefirmato per un caricamento, quindi carica un file (richiesta PUT).

Importazioni.

```
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;
```

Genera l'URL.

```
   /* Create a presigned URL to use in a subsequent PUT request */
   public String createPresignedUrl(String bucketName, String keyName,
Map<String, String> metadata) {
       try (S3Presigner presigner = S3Presigner.create()) {

           PutObjectRequest objectRequest = PutObjectRequest.builder()
                   .bucket(bucketName)
                   .key(keyName)
                   .metadata(metadata)
                   .build();

           PutObjectPresignRequest presignRequest =
PutObjectPresignRequest.builder()
                   .signatureDuration(Duration.ofMinutes(10))  // The URL
expires in 10 minutes.
                   .putObjectRequest(objectRequest)
                   .build();


           PresignedPutObjectRequest presignedRequest =
presigner.presignPutObject(presignRequest);
```

```
            String myURL = presignedRequest.url().toString();
            logger.info("Presigned URL to upload a file to: [{}]", myURL);
            logger.info("HTTP method: [{}]",
 presignedRequest.httpRequest().method());

            return presignedRequest.url().toExternalForm();
        }
    }
```

Caricate un oggetto file utilizzando uno dei tre approcci seguenti.

Utilizzate la classe JDK `HttpURLConnection` (dalla v1.1) per eseguire il caricamento.

```
    /* 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);
        }
    }
```

Utilizzate la classe JDK `HttpClient` (dalla v11) per eseguire il caricamento.

```
    /* 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);
        }
    }
```

Utilizzate la `SdkHttpClient` classe AWS for Java V2 per eseguire il caricamento.

```
    /* 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 per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea un URL prefirmato per caricare un oggetto in 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);
```

```
  }
};
```

Crea un URL prefirmato per scaricare un oggetto da 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);
  }
};
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea una richiesta prefirmata `GetObject` e usa l'URL per scaricare un oggetto.

```
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
}
```

Crea una richiesta `GetObject` predefinita con opzioni avanzate.

```
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
}
```

Crea una richiesta prefirmata `PutObject` e usala per caricare un oggetto.

```
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)
}
```

- Per ulteriori informazioni, consulta la Guida per gli sviluppatori di AWS SDK per Swift.

PHP

SDK per PHP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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 per Python (Boto3)

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Genera un URL prefirmato in grado di eseguire un'operazione S3 per un periodo di tempo limitato. Utilizza il pacchetto Requests per effettuare una richiesta con l'URL.

```
import argparse
import logging
import boto3
from botocore.exceptions import ClientError
import requests

logger = logging.getLogger(__name__)
```

```
def generate_presigned_url(s3_client, client_method, method_parameters,
 expires_in):
    """
    Generate a presigned Amazon S3 URL that can be used to perform an action.

    :param s3_client: A Boto3 Amazon S3 client.
    :param client_method: The name of the client method that the URL performs.
    :param method_parameters: The parameters of the specified client method.
    :param expires_in: The number of seconds the presigned URL is valid for.
    :return: The presigned URL.
    """
    try:
        url = s3_client.generate_presigned_url(
            ClientMethod=client_method, Params=method_parameters,
 ExpiresIn=expires_in
        )
        logger.info("Got presigned URL: %s", url)
    except ClientError:
        logger.exception(
            "Couldn't get a presigned URL for client method '%s'.", client_method
        )
        raise
    return url


def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Welcome to the Amazon S3 presigned URL demo.")
    print("-" * 88)

    parser = argparse.ArgumentParser()
    parser.add_argument("bucket", help="The name of the bucket.")
    parser.add_argument(
        "key",
        help="For a GET operation, the key of the object in Amazon S3. For a "
        "PUT operation, the name of a file to upload.",
    )
    parser.add_argument("action", choices=("get", "put"), help="The action to
 perform.")
    args = parser.parse_args()
```

```
    s3_client = boto3.client("s3")
    client_action = "get_object" if args.action == "get" else "put_object"
    url = generate_presigned_url(
        s3_client, client_action, {"Bucket": args.bucket, "Key": args.key}, 1000
    )

    print("Using the Requests package to send a request to the URL.")
    response = None
    if args.action == "get":
        response = requests.get(url)
    elif args.action == "put":
        print("Putting data to the URL.")
        try:
            with open(args.key, "r") as object_file:
                object_text = object_file.read()
            response = requests.put(url, data=object_text)
        except FileNotFoundError:
            print(
                f"Couldn't find {args.key}. For a PUT operation, the key must be
 the "
                f"name of a file that exists on your computer."
            )

    if response is not None:
        print("Got response:")
        print(f"Status: {response.status_code}")
        print(response.text)

    print("-" * 88)


if __name__ == "__main__":
    usage_demo()
```

Genera una richiesta POST prefirmata per caricare un file.

```
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 per Ruby

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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 per Rust

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea richieste di preassegnazione per oggetti GET e 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(())
}
```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Una pagina Web che elenca gli oggetti Amazon S3 utilizzando un SDK AWS

Il codice di esempio seguente mostra come elencare gli oggetti Amazon S3 in una pagina Web.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Il codice seguente è il componente React pertinente che effettua chiamate all' AWS SDK. Una versione eseguibile dell'applicazione contenente questo componente è disponibile al link precedente. 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;
```

- Per i dettagli sull'API, consulta la sezione API ListObjectsReference AWS SDK for JavaScript .

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Eliminare caricamenti multiparte incompleti su Amazon S3 utilizzando un SDK AWS

Il seguente esempio di codice mostra come eliminare o interrompere i caricamenti multiparte incompleti di Amazon S3.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Per interrompere i caricamenti in più parti che sono in corso o incompleti per qualsiasi motivo, puoi creare un elenco di caricamenti e quindi eliminarli come mostrato nell'esempio seguente.

```
   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);
           }
       }
   }
```

Per eliminare i caricamenti incompleti in più parti iniziati prima o dopo una data, puoi eliminare selettivamente i caricamenti in più parti in base a un momento temporale, come mostrato nell'esempio seguente.

```
   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);
                }
            }
        }
    }
```

Se hai accesso all'ID di caricamento dopo aver iniziato un caricamento in più parti, puoi eliminare il caricamento in corso utilizzando l'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);
```

```
        }
    }
```

Per eliminare in modo coerente i caricamenti incompleti in più parti più vecchi di un certo numero di giorni, imposta una configurazione del ciclo di vita del bucket per il bucket. L'esempio seguente mostra come creare una regola per eliminare i caricamenti incompleti più vecchi di 7 giorni.

```
   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());
       }
   }
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .
  - AbortMultipartUpload
  - ListMultipartUploads
  - PutBucketLifecycleConfiguration

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scaricare tutti gli oggetti da un bucket Amazon Simple Storage Service (Amazon S3) in una directory locale

L'esempio di codice seguente mostra come scaricare tutti gli oggetti da un bucket Amazon Simple Storage Service (Amazon S3) in una directory locale.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Usa un S3 TransferManager per scaricare tutti gli oggetti S3 nello stesso bucket S3. Visualizza il file completo ed esegui il test.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.services.s3.model.ObjectIdentifier;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedDirectoryDownload;
import software.amazon.awssdk.transfer.s3.model.DirectoryDownload;
import software.amazon.awssdk.transfer.s3.model.DownloadDirectoryRequest;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.HashSet;
import java.util.Set;
import java.util.UUID;
import java.util.stream.Collectors;
```

```
   public Integer downloadObjectsToDirectory(S3TransferManager transferManager,
           URI destinationPathURI, String bucketName) {
       DirectoryDownload directoryDownload =
transferManager.downloadDirectory(DownloadDirectoryRequest.builder()
               .destination(Paths.get(destinationPathURI))
               .bucket(bucketName)
               .build());
       CompletedDirectoryDownload completedDirectoryDownload =
directoryDownload.completionFuture().join();

       completedDirectoryDownload.failedTransfers()
               .forEach(fail -> logger.warn("Object [{}] failed to transfer",
fail.toString()));
       return completedDirectoryDownload.failedTransfers().size();
   }
```

- Per i dettagli sull'API, consulta DownloadDirectoryla sezione API Reference.AWS SDK for Java 2.x

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Ottieni un oggetto Amazon S3 da un punto di accesso multiregionale utilizzando un SDK AWS

Il seguente esempio di codice mostra come ottenere un oggetto da un punto di accesso multiregionale.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Configura il client S3 per utilizzare l'algoritmo di firma Asymmetric 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
        }
```

Utilizzate l'ARN del punto di accesso multiregionale anziché il nome di un bucket per recuperare l'oggetto.

```
    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
    }
```

- Per ulteriori informazioni, consulta la Guida per gli sviluppatori di AWS SDK per Swift.
- Per i dettagli sull'API, consulta il riferimento GetObjectall'API AWS SDK for Kotlin.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Ottieni un oggetto da un bucket Amazon S3 utilizzando un AWS SDK, specificando un'intestazione If-Modified-Since

L'esempio di codice seguente mostra come leggere i dati da un oggetto in un bucket S3, ma solo se il bucket non è stato modificato dall'ultimo recupero.

Rust

SDK per Rust

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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(())
}
```

- Per i dettagli sulle API, consulta la GetObjectguida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Inizia a usare bucket e oggetti Amazon S3 utilizzando un SDK AWS

Gli esempi di codice seguenti mostrano come:

- Crea un bucket e carica un file in tale bucket.
- Scaricare un oggetto da un bucket.
- Copiare un oggetto in una sottocartella in un bucket.
- Elencare gli oggetti in un bucket.
- Elimina il bucket e tutti gli oggetti in esso contenuti.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for .NET .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject

- ListObjectsV2
- PutObject

Bash

AWS CLI con script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

Le funzioni di Amazon S3 utilizzate in questo scenario.

```
###############################################################################
# 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
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento dei comandi AWS CLI .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

C++

SDK per C++

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for C++ .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Go

SDK per Go V2

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Definisci una struttura che racchiude le azioni del bucket e dell'oggetto utilizzate dallo scenario.

```
// 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
}
```

Esegui uno scenario interattivo che ti mostri come utilizzare i bucket e gli oggetti 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))
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Go .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Java

SDK per Java 2.x

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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 "";
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2

- PutObject

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Innanzitutto, importa tutti i moduli necessari.

```
// 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";
```

Le importazioni precedenti fanno riferimento ad alcune utilità di supporto. Queste utilità sono locali al GitHub repository collegato all'inizio di questa sezione. Come riferimento, consulta le implementazioni di tali utilità riportate di seguito.

```
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`;
};
```

Gli oggetti in S3 sono archiviati in "bucket". Definiamo una funzione per creare un nuovo 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;
};
```

I bucket contengono "oggetti". Questa funzione carica il contenuto di una directory nel bucket come oggetti.

```
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.`);
  }
};
```

Dopo aver caricato gli oggetti, verifica che siano stati caricati correttamente. Puoi usare ListObjects per questo. Utilizzerai la proprietà 'Key', ma ci sono anche altre proprietà utili nella risposta.

```
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 volte potresti voler copiare un oggetto da un bucket ad altri bucket. Usa il CopyObject comando per questo.

```
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();
  }
};
```

Non esiste un metodo SDK per ottenere più oggetti da un bucket. Creerai invece un elenco di oggetti da scaricare ed eseguirai iterazioni su di essi.

```
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");
```

```
};
```

È il momento di eseguire una pulizia delle risorse. Prima di poter essere eliminato, un bucket deve essere vuoto. Queste due funzioni svuotano ed eliminano il 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 funzione "principale" esegue entrambe le operazioni. Se esegui direttamente questo file, verrà chiamata la funzione principale.

```
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);
  }
};
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for JavaScript .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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!")
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Kotlin.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

PHP

SDK per PHP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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";
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for PHP .
  - CopyObject
  - CreateBucket

- DeleteBucket
- DeleteObjects
- GetObject
- ListObjectsV2
- PutObject

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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"))
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Ruby

SDK per Ruby

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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__
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Ruby .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Codice per la crate binario che esegue lo scenario.

```
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(())
}
```

Una crate libreria con operazioni comuni chiamate dal 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
}
```

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Rust.
  - CopyObject
  - CreateBucket

- DeleteBucket
- DeleteObjects
- GetObject
- ListObjectsV2
- PutObject

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.
```

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per SAP ABAP.
  - CopyObject
  - CreateBucket

- DeleteBucket
- DeleteObjects
- GetObject
- ListObjectsV2
- PutObject

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Una classe Swift che gestisce le chiamate all'SDK per Swift.

```
import Foundation
import AWSS3
import ClientRuntime
import AWSClientRuntime

/// A class containing all the code that interacts with the AWS SDK for Swift.
public class ServiceHandler {
    let client: S3Client

    /// Initialize and return a new ``ServiceHandler`` object, which is used to
 drive the AWS calls
    /// used for the example.
    ///
    /// - Returns: A new ``ServiceHandler`` object, ready to be called to
    ///            execute AWS operations.
```

```
    public init() async {
        do {
            client = try S3Client(region: "us-east-2")
        } catch {
            print("ERROR: ", dump(error, name: "Initializing S3 client"))
            exit(1)
        }
    }

    /// Create a new user given the specified name.
    ///
    /// - Parameters:
    ///   - name: Name of the bucket to create.
    /// Throws an exception if an error occurs.
    public func createBucket(name: String) async throws {
        let config = S3ClientTypes.CreateBucketConfiguration(
            locationConstraint: .usEast2
        )
        let input = CreateBucketInput(
            bucket: name,
            createBucketConfiguration: config
        )
        _ = try await client.createBucket(input: input)
    }

    /// Delete a bucket.
    /// - Parameter name: Name of the bucket to delete.
    public func deleteBucket(name: String) async throws {
        let input = DeleteBucketInput(
            bucket: name
        )
        _ = try await client.deleteBucket(input: input)
    }

    /// Upload a file from local storage to the bucket.
    /// - Parameters:
    ///   - bucket: Name of the bucket to upload the file to.
    ///   - key: Name of the file to create.
    ///   - file: Path name of the file to upload.
    public func uploadFile(bucket: String, key: String, file: String) async
throws {
        let fileUrl = URL(fileURLWithPath: file)
        let fileData = try Data(contentsOf: fileUrl)
        let dataStream = ByteStream.from(data: fileData)
```

```
        let input = PutObjectInput(
            body: dataStream,
            bucket: bucket,
            key: key
        )
        _ = try await client.putObject(input: input)
    }

    /// Create a file in the specified bucket with the given name. The new
    /// file's contents are uploaded from a `Data` object.
    ///
    /// - Parameters:
    ///   - bucket: Name of the bucket to create a file in.
    ///   - key: Name of the file to create.
    ///   - data: A `Data` object to write into the new file.
    public func createFile(bucket: String, key: String, withData data: Data)
async throws {
        let dataStream = ByteStream.from(data: data)

        let input = PutObjectInput(
            body: dataStream,
            bucket: bucket,
            key: key
        )
        _ = try await client.putObject(input: input)
    }

    /// Download the named file to the given directory on the local device.
    ///
    /// - Parameters:
    ///   - bucket: Name of the bucket that contains the file to be copied.
    ///   - key: The name of the file to copy from the bucket.
    ///   - to: The path of the directory on the local device where you want to
    ///     download the file.
    public func downloadFile(bucket: String, key: String, to: String) async
throws {
        let fileUrl = URL(fileURLWithPath: to).appendingPathComponent(key)

        let input = GetObjectInput(
            bucket: bucket,
            key: key
        )
        let output = try await client.getObject(input: input)
```

```
        // Get the data stream object. Return immediately if there isn't one.
        guard let body = output.body,
              let data = try await body.readData() else {
            return
        }
        try data.write(to: fileUrl)
    }

    /// Read the specified file from the given S3 bucket into a Swift
    /// `Data` object.
    ///
    /// - Parameters:
    ///   - bucket: Name of the bucket containing the file to read.
    ///   - key: Name of the file within the bucket to read.
    ///
    /// - Returns: A `Data` object containing the complete file data.
    public func readFile(bucket: String, key: String) async throws -> Data {
        let input = GetObjectInput(
            bucket: bucket,
            key: key
        )
        let output = try await client.getObject(input: input)

        // Get the stream and return its contents in a `Data` object. If
        // there is no stream, return an empty `Data` object instead.
        guard let body = output.body,
              let data = try await body.readData() else {
            return "".data(using: .utf8)!
        }

        return data
    }

    /// Copy a file from one bucket to another.
    ///
    /// - Parameters:
    ///   - sourceBucket: Name of the bucket containing the source file.
    ///   - name: Name of the source file.
    ///   - destBucket: Name of the bucket to copy the file into.
    public func copyFile(from sourceBucket: String, name: String, to destBucket:
 String) async throws {
        let srcUrl = ("\(sourceBucket)/
\(name)").addingPercentEncoding(withAllowedCharacters: .urlPathAllowed)
```

```
        let input = CopyObjectInput(
            bucket: destBucket,
            copySource: srcUrl,
            key: name
        )
        _ = try await client.copyObject(input: input)
    }

    /// Deletes the specified file from Amazon S3.
    ///
    /// - Parameters:
    ///   - bucket: Name of the bucket containing the file to delete.
    ///   - key: Name of the file to delete.
    ///
    public func deleteFile(bucket: String, key: String) async throws {
        let input = DeleteObjectInput(
            bucket: bucket,
            key: key
        )

        do {
            _ = try await client.deleteObject(input: input)
        } catch {
            throw error
        }
    }

    /// Returns an array of strings, each naming one file in the
    /// specified bucket.
    ///
    /// - Parameter bucket: Name of the bucket to get a file listing for.
    /// - Returns: An array of `String` objects, each giving the name of
    ///            one file contained in the bucket.
    public func listBucketFiles(bucket: String) async throws -> [String] {
        let input = ListObjectsV2Input(
            bucket: bucket
        )
        let output = try await client.listObjectsV2(input: input)
        var names: [String] = []

        guard let objList = output.contents else {
            return []
        }
```

```
        for obj in objList {
            if let objName = obj.key {
                names.append(objName)
            }
        }

        return names
    }
}
```

Un programma Swift a riga di comando per gestire le chiamate 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)
        }
    }
}
```

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Swift.
  - CopyObject

- CreateBucket
- DeleteBucket
- DeleteObjects
- GetObject
- ListObjectsV2
- PutObject

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Inizia a utilizzare la crittografia per oggetti Amazon S3 utilizzando un SDK AWS

L'esempio di codice seguente mostra come iniziare a utilizzare la crittografia per gli oggetti Amazon S3.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for .NET .
  - CopyObject
  - GetObject
  - GetObjectMetadata

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Inizia a usare i tag per gli oggetti Amazon S3 utilizzando un SDK AWS

L'esempio di codice seguente mostra come utilizzare i tag con gli oggetti Amazon S3.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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}'");
            }
        }
    }
```

- Per i dettagli sull'API, GetObjectTaggingconsulta AWS SDK for .NET API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Ottieni la configurazione di conservazione legale di un oggetto Amazon S3 utilizzando un SDK AWS

I seguenti esempi di codice mostrano come ottenere la configurazione di conservazione legale di un bucket S3.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
        }
    }
```

- Per i dettagli sull'API, GetObjectLegalHoldconsulta AWS SDK for .NET API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
   }
```

- Per i dettagli sull'API, GetObjectLegalHoldconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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");
```

```
}
```

- Per i dettagli sull'API, GetObjectLegalHoldconsulta AWS SDK for JavaScript API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Lavora con le funzionalità di blocco degli oggetti di Amazon S3 utilizzando un SDK AWS

I seguenti esempi di codice mostrano come utilizzare le funzionalità di blocco degli oggetti di S3.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui uno scenario interattivo che dimostri le funzionalità di blocco degli oggetti di 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 classe wrapper per le funzioni 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;
        }
```

```
    }

}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for .NET .
  - GetObjectLegalHold
  - GetObjectLockConfiguration
  - GetObjectRetention
  - PutObjectLegalHold
  - PutObjectLockConfiguration
  - PutObjectRetention

Go

SDK per Go V2

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui uno scenario interattivo che dimostri le funzionalità di blocco degli oggetti di 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))
}
```

Definisci una struttura che racchiuda le azioni S3 utilizzate in questo esempio.

```
// 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
}
```

Pulisci le risorse.

```
// 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)
  }
 }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Go .
  - GetObjectLegalHold
  - GetObjectLockConfiguration
  - GetObjectRetention
  - PutObjectLegalHold
  - PutObjectLockConfiguration
  - PutObjectRetention

Java

SDK per Java 2.x

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui uno scenario interattivo che dimostri le funzionalità di blocco degli oggetti di 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 classe wrapper per le funzioni 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());
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .
  - GetObjectLegalHold
  - GetObjectLockConfiguration
  - GetObjectRetention
  - PutObjectLegalHold
  - PutObjectLockConfiguration
  - PutObjectRetention

JavaScript

SDK per (v3) JavaScript

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

index.js- Punto di ingresso per il flusso di lavoro. Questo orchestra tutti i passaggi. Visita GitHub per vedere i dettagli di implementazione di Scenario, ScenarioInput ScenarioOutput, e. 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- Invia messaggi di benvenuto alla console.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
/**
 * @typedef {import("@aws-doc-sdk-examples/lib/scenario/index.js")} Scenarios
 */

/**
 * @param {Scenarios} scenarios
 */
const welcome = (scenarios) =>
  new scenarios.ScenarioOutput(
    "welcome",
    `Welcome to the Amazon Simple Storage Service (S3) Object Locking Workflow
 Scenario. For this workflow, we will use the AWS SDK for JavaScript to create
 several S3 buckets and files to demonstrate working with S3 locking features.`,
    { header: true },
  );

/**
 * @param {Scenarios} scenarios
 */
const welcomeContinue = (scenarios) =>
  new scenarios.ScenarioInput(
    "welcomeContinue",
    "Press Enter when you are ready to start.",
    { type: "confirm" },
  );

export { welcome, welcomeContinue };
```

setup.steps.js- Distribuisci bucket, oggetti e impostazioni dei file.

```
// 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- Visualizza ed elimina i file nei bucket.

```
// 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- Distruggi tutte le risorse create.

```
// 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 };
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for JavaScript .
  - GetObjectLegalHold
  - GetObjectLockConfiguration
  - GetObjectRetention
  - PutObjectLegalHold
  - PutObjectLockConfiguration
  - PutObjectRetention

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Gestisci gli elenchi di controllo degli accessi (ACL) per i bucket Amazon S3 utilizzando un SDK AWS

Gli esempi di codice seguenti mostrano come gestire le liste di controllo degli accessi (ACL) per i bucket Amazon S3.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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,
            });
        }

    }
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for .NET .
  - GetBucketAcl
  - GetObjectAcl
  - PutBucketAcl
  - PutObjectAcl

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Gestisci oggetti Amazon S3 con versioni in batch con una funzione Lambda utilizzando un SDK AWS

L'esempio di codice seguente mostra come gestire gli oggetti con versione S3 in batch con una funzione Lambda.

Python

SDK per Python (Boto3)

Mostra come manipolare oggetti con versione di Amazon Simple Storage Service (Amazon S3) in batch creando processi che richiamano funzioni per eseguire l'elaborazione. AWS Lambda Questo esempio mostra come creare un bucket abilitato per le versioni, caricare le strofe dalla poesia You Are Old, Father William di Lewis Carroll e utilizzare i processi batch Amazon S3 per eseguire varie operazioni sulla poesia.

Scopri come:

- Creare funzioni Lambda che operano su oggetti con versione.
- Creare un manifesto di oggetti da aggiornare.
- Creare processi batch che richiamano le funzioni Lambda per aggiornare gli oggetti.
- Eliminare funzioni Lambda.
- Svuotare ed eliminare un bucket con versione.

Questo esempio è visualizzato al meglio su. GitHub Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, vedi l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- Amazon S3

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Analizza gli URI di Amazon S3 utilizzando un SDK AWS

L'esempio di codice seguente mostra come analizzare gli URI di Amazon S3 per estrarre componenti importanti, come il nome del bucket e la chiave dell’oggetto.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Analizza un URI Amazon S3 utilizzando la classe S3Uri.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.S3Uri;
import software.amazon.awssdk.services.s3.S3Utilities;

import java.net.URI;
import java.util.List;
import java.util.Map;

    /**
     *
     * @param s3Client    - An S3Client through which you acquire an S3Uri
 instance.
     * @param s3ObjectUrl - A complex URL (String) that is used to demonstrate
 S3Uri
     *                    capabilities.
     */
    public static void parseS3UriExample(S3Client s3Client, String s3ObjectUrl) {
        logger.info(s3ObjectUrl);
        // Console output:
        // 'https://s3.us-west-1.amazonaws.com/myBucket/resources/doc.txt?
versionId=abc123&partNumber=77&partNumber=88'.

        // Create an S3Utilities object using the configuration of the s3Client.
        S3Utilities s3Utilities = s3Client.utilities();

        // From a String URL create a URI object to pass to the parseUri()
 method.
        URI uri = URI.create(s3ObjectUrl);
```

```
        S3Uri s3Uri = s3Utilities.parseUri(uri);

        // If the URI contains no value for the Region, bucket or key, the SDK
returns
        // an empty Optional.
        // The SDK returns decoded URI values.

        Region region = s3Uri.region().orElse(null);
        log("region", region);
        // Console output: 'region: us-west-1'.

        String bucket = s3Uri.bucket().orElse(null);
        log("bucket", bucket);
        // Console output: 'bucket: myBucket'.

        String key = s3Uri.key().orElse(null);
        log("key", key);
        // Console output: 'key: resources/doc.txt'.

        Boolean isPathStyle = s3Uri.isPathStyle();
        log("isPathStyle", isPathStyle);
        // Console output: 'isPathStyle: true'.

        // If the URI contains no query parameters, the SDK returns an empty map.
        Map<String, List<String>> queryParams = s3Uri.rawQueryParameters();
        log("rawQueryParameters", queryParams);
        // Console output: 'rawQueryParameters: {versionId=[abc123],
partNumber=[77,
        // 88]}'.

        // Retrieve the first or all values for a query parameter as shown in the
        // following code.
        String versionId =
s3Uri.firstMatchingRawQueryParameter("versionId").orElse(null);
        log("firstMatchingRawQueryParameter-versionId", versionId);
        // Console output: 'firstMatchingRawQueryParameter-versionId: abc123'.

        String partNumber =
s3Uri.firstMatchingRawQueryParameter("partNumber").orElse(null);
        log("firstMatchingRawQueryParameter-partNumber", partNumber);
        // Console output: 'firstMatchingRawQueryParameter-partNumber: 77'.

        List<String> partNumbers =
s3Uri.firstMatchingRawQueryParameters("partNumber");
```

```
        log("firstMatchingRawQueryParameter", partNumbers);
        // Console output: 'firstMatchingRawQueryParameter: [77, 88]'.

        /*
         * Object keys and query parameters with reserved or unsafe characters,
 must be
         * URL-encoded.
         * For example replace whitespace " " with "%20".
         * Valid:
         * "https://s3.us-west-1.amazonaws.com/myBucket/object%20key?query=
%5Bbrackets%5D"
         * Invalid:
         * "https://s3.us-west-1.amazonaws.com/myBucket/object key?
query=[brackets]"
         *
         * Virtual-hosted-style URIs with bucket names that contain a dot, ".",
 the dot
         * must not be URL-encoded.
         * Valid: "https://my.Bucket.s3.us-west-1.amazonaws.com/key"
         * Invalid: "https://my%2EBucket.s3.us-west-1.amazonaws.com/key"
         */
    }

    private static void log(String s3UriElement, Object element) {
        if (element == null) {
            logger.info("{}: {}", s3UriElement, "null");
        } else {
            logger.info("{}: {}", s3UriElement, element);
        }
    }
```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esegui una copia multiparte di un oggetto Amazon S3 utilizzando un SDK AWS

L'esempio di codice seguente mostra come eseguire una copia in più parti di un oggetto Amazon S3.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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");
            }
        }
    }
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for .NET .
  - CompleteMultipartUpload
  - CreateMultipartUpload
  - GetObjectMetadata
  - UploadPartCopy

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esegui un caricamento multiparte di un oggetto Amazon S3 utilizzando un SDK AWS

L'esempio di codice seguente mostra come eseguire un caricamento in più parti in un oggetto Amazon S3.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Gli esempi di codice utilizzano le seguenti importazioni.

```
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;
```

Utilizza S3 Transfer Manager sul client S3 basato su CRT AWS per eseguire in modo trasparente un caricamento in più parti quando le dimensioni del contenuto superano una soglia. Le dimensioni soglia predefinite sono di 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();
    }
```

Usa l'API S3Client per eseguire un caricamento in più parti.

```
    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()));
    }
```

Utilizza l'AsyncClient API S3 con il supporto multiparte abilitato per eseguire un caricamento in più parti.

```
    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.");
    }
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .
  - CompleteMultipartUpload
  - CreateMultipartUpload
  - UploadPart

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Tieni traccia del caricamento o del download di un oggetto Amazon S3 utilizzando un SDK AWS

Il seguente esempio di codice mostra come tenere traccia del caricamento o del download di un oggetto Amazon S3.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Tieni traccia dello stato di avanzamento del caricamento di un file.

```
   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!
       */
   }
```

Tieni traccia dello stato di avanzamento del download di un file.

```
   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!
        */
    }
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .
  - GetObject
  - PutObject

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di approcci per i test di unità e integrazione con un SDK AWS

Il seguente esempio di codice mostra esempi di tecniche ottimali per la scrittura di test unitari e di integrazione utilizzando un AWS SDK.

Rust

SDK per Rust

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Cargo.toml per esempi di test.

```
[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"
```

Esempio di test di unità utilizzando automock e un wrapper di servizi.

```
// 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]
```

Esempio di test di integrazione utilizzando 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]
```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Caricare in modo ricorsivo una directory locale in un bucket Amazon Simple Storage Service (Amazon S3)

L'esempio di codice seguente mostra come caricare una directory locale in modo ricorsivo in un bucket Amazon Simple Storage Service (Amazon S3).

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Usa un S3 TransferManager per caricare una directory locale. Visualizza il file completo ed esegui il test.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.s3.model.ObjectIdentifier;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedDirectoryUpload;
import software.amazon.awssdk.transfer.s3.model.DirectoryUpload;
import software.amazon.awssdk.transfer.s3.model.UploadDirectoryRequest;

import java.net.URI;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.file.Paths;
import java.util.UUID;

    public Integer uploadDirectory(S3TransferManager transferManager,
            URI sourceDirectory, String bucketName) {
        DirectoryUpload directoryUpload =
 transferManager.uploadDirectory(UploadDirectoryRequest.builder()
                .source(Paths.get(sourceDirectory))
                .bucket(bucketName)
                .build());
```

```
        CompletedDirectoryUpload completedDirectoryUpload =
directoryUpload.completionFuture().join();
        completedDirectoryUpload.failedTransfers()
                .forEach(fail -> logger.warn("Object [{}] failed to transfer",
fail.toString()));
        return completedDirectoryUpload.failedTransfers().size();
    }
```

- Per i dettagli sull'API, consulta la sezione UploadDirectory AWS SDK for Java 2.xAPI Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Carica o scarica file di grandi dimensioni da e verso Amazon S3 utilizzando un SDK AWS

Gli esempi di codice seguente mostrano come caricare o scaricare file di grandi dimensioni in e da Amazon S3.

Per ulteriori informazioni, consulta Caricamento di un oggetto utilizzando il caricamento in più parti.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Chiama le funzioni che trasferiscono file da e verso un bucket S3 utilizzando 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);
}
```

Caricamento di un singolo file.

```
        /// <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;
            }
        }
```

Caricamento di un'intera directory locale.

```
        /// <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;
```

```
            }
        }
```

Download di un singolo file.

```
        /// <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}"));
        }
```

Download dei contenuti di un bucket 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;
        }
```

Tieni traccia dello stato di avanzamento di un caricamento utilizzando. 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}");
        }
    }
```

Carica un oggetto con la crittografia.

```
    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 per Go V2

> **Note**
>
> C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Carica un oggetto di grandi dimensioni utilizzando un gestore di caricamento per suddividere i dati in parti e caricarli contemporaneamente.

```
// 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
}
```

Scarica un oggetto di grandi dimensioni utilizzando un gestore di download per ottenere i dati in parti e scaricarli contemporaneamente.

```
// 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 per Java 2.x

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Richiama le funzioni che trasferiscono file da e verso un bucket S3 utilizzando S3. TransferManager

```
   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();
   }
```

Caricamento di un'intera directory locale.

```
   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();
   }
```

Caricamento di un singolo file.

```
    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 per (v3) JavaScript

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Carica un file di grandi dimensioni.

```
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);
    }
  }
};
```

Scarica un file di grandi dimensioni.

```
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 per Python (Boto3)

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea funzioni che trasferiscono i file utilizzando diverse impostazioni disponibili del gestore di trasferimento. Utilizza una classe di callback per scrivere l'avanzamento del callback durante il trasferimento dei file.

```
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
```

Esegui le funzioni del gestore di trasferimento e ottieni i risultati.

```
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 per Rust

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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(())
}
```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Carica uno stream di dimensioni sconosciute su un oggetto Amazon S3 utilizzando un SDK AWS

Gli esempi di codice seguenti mostrano come caricare un flusso di dimensioni sconosciute in un oggetto Amazon S3.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Usa il Client S3 basato su CRT AWS.

```
import com.example.s3.util.AsyncExampleUtils;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.async.AsyncRequestBody;
import software.amazon.awssdk.core.async.BlockingInputStreamAsyncRequestBody;
import software.amazon.awssdk.core.exception.SdkException;
import software.amazon.awssdk.services.s3.S3AsyncClient;
import software.amazon.awssdk.services.s3.model.PutObjectResponse;

import java.io.ByteArrayInputStream;
import java.util.UUID;
import java.util.concurrent.CompletableFuture;

    /**
     * @param s33CrtAsyncClient - To upload content from a stream of unknown
 size, use the AWS CRT-based S3 client. For more information, see
     *                          https://docs.aws.amazon.com/sdk-for-java/latest/
developer-guide/crt-based-s3-client.html.
     * @param bucketName - The name of the bucket.
     * @param key - The name of the object.
     * @return software.amazon.awssdk.services.s3.model.PutObjectResponse -
 Returns metadata pertaining to the put object operation.
     */
    public PutObjectResponse putObjectFromStream(S3AsyncClient s33CrtAsyncClient,
 String bucketName, String key) {

        BlockingInputStreamAsyncRequestBody body =
                AsyncRequestBody.forBlockingInputStream(null); // 'null'
 indicates a stream will be provided later.

        CompletableFuture<PutObjectResponse> responseFuture =
```

```
                s33CrtAsyncClient.putObject(r -> r.bucket(bucketName).key(key),
 body);

        // AsyncExampleUtils.randomString() returns a random string up to 100
 characters.
        String randomString = AsyncExampleUtils.randomString();
        logger.info("random string to upload: {}: length={}", randomString,
 randomString.length());

        // Provide the stream of data to be uploaded.
        body.writeInputStream(new ByteArrayInputStream(randomString.getBytes()));

        PutObjectResponse response = responseFuture.join(); // Wait for the
 response.
        logger.info("Object {} uploaded to bucket {}.", key, bucketName);
        return response;
    }
}
```

Usa Amazon S3 Transfer Manager.

```
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();
    }
}
```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usa i checksum per lavorare con un oggetto Amazon S3 utilizzando un SDK AWS

L'esempio di codice seguente mostra come utilizzare i checksum per lavorare con un oggetto Amazon S3.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Gli esempi di codice utilizzano un sottoinsieme delle seguenti importazioni.

```
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;
```

Specifica un algoritmo di checksum per il metodo `putObject` quando crei il `PutObjectRequest`.

```
    public void putObjectWithChecksum() {
        s3Client.putObject(b -> b
                .bucket(bucketName)
                .key(key)
                .checksumAlgorithm(ChecksumAlgorithm.CRC32),
                RequestBody.fromString("This is a test"));
    }
```

Verifica il checksum per il `getObject` metodo quando crei il GetObjectRequest.

```
    public GetObjectResponse getObjectWithChecksum() {
        return s3Client.getObject(b -> b
                .bucket(bucketName)
                .key(key)
                .checksumMode(ChecksumMode.ENABLED))
                .response();
    }
```

Precalcola un checksum per il metodo `putObject` quando crei il `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)));
    }
```

Utilizza S3 Transfer Manager sul client S3 basato su CRT AWS per eseguire in modo trasparente un caricamento in più parti quando le dimensioni del contenuto superano una soglia. Le dimensioni soglia predefinite sono di 8 MB.

Puoi specificare un algoritmo di checksum da utilizzare nell'SDK. Per impostazione predefinita, l'SDK utilizza l'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();
    }
```

Utilizza l'API S3Client o (API S3) per eseguire un AsyncClient caricamento in più parti. Se specifichi un checksum aggiuntivo, devi specificare l'algoritmo da utilizzare all'avvio del caricamento. Inoltre, devi specificare l'algoritmo per ogni richiesta parte e fornire il checksum calcolato per ciascuna parte dopo che è stata caricata.

```
    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()));
   }
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .
  - CompleteMultipartUpload
  - CreateMultipartUpload
  - UploadPart

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Lavora con oggetti con versione di Amazon S3 utilizzando un SDK AWS

L'esempio di codice seguente mostra come:

- Creazione un bucket S3 con versione.
- Ottenimento di tutte le versioni di un oggetto.
- Ripristino di un oggetto a una versione precedente.
- Eliminazione e ripristino di un oggetto con versione.
- Eliminazione permanente di tutte le versioni di un oggetto

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Creazione di funzioni che eseguono il wrap delle operazioni 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
```

Caricamento la strofa di una poesia in un oggetto con versione ed esecuzione di una serie di operazioni su di esso.

```
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!")
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
  - CreateBucket
  - DeleteObject
  - ListObjectVersions
  - PutBucketLifecycleConfiguration

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi serverless per Amazon S3 che utilizzano SDK AWS

I seguenti esempi di codice mostrano come usare Amazon S3 con AWS SDK.

Esempi

- Richiamo di una funzione Lambda da un trigger Amazon S3

Richiamo di una funzione Lambda da un trigger Amazon S3

I seguenti esempi di codice mostrano come implementare una funzione Lambda che riceve un evento attivato dal caricamento di un oggetto in un bucket S3. La funzione recupera il nome del bucket S3 e la chiave dell'oggetto dal parametro evento e chiama l'API Amazon S3 per recuperare e registrare il tipo di contenuto dell'oggetto.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository di Esempi serverless.

Utilizzo di un evento S3 con Lambda tramite .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 per Go V2

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository di Esempi serverless.

Utilizzo di un evento S3 con Lambda tramite 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 per Java 2.x

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository di Esempi serverless.

Utilizzo di un evento S3 con Lambda tramite 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 per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository di Esempi serverless.

Consumo di un evento S3 con JavaScript Lambda utilizzando.

```
// 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);
    }
};
```

Consumo di un evento S3 con TypeScript Lambda utilizzando.

```
// 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 per PHP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository di Esempi serverless.

Consumo di un evento S3 con Lambda utilizzando 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 per Python (Boto3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository di Esempi serverless.

Utilizzo di un evento S3 con Lambda tramite 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 per Ruby

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository di Esempi serverless.

Consumo di un evento S3 con Lambda utilizzando 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 per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository di Esempi serverless.

Utilizzo di un evento S3 con Lambda tramite 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(())
}
```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di servizi multipli per Amazon S3 che utilizzano SDK AWS

Le seguenti applicazioni di esempio utilizzano AWS gli SDK per combinare Amazon S3 con altri. Servizi AWS Ogni esempio include un collegamento a GitHub, dove puoi trovare istruzioni su come configurare ed eseguire l'applicazione.

Esempi

- Creazione di un'app Amazon Transcribe
- Convertire testo in voce e viceversa utilizzando un AWS SDK
- Creazione di un'applicazione di gestione delle risorse fotografiche che consente agli utenti di gestire le foto utilizzando etichette
- Creazione di un'applicazione Amazon Textract explorer
- Rileva i DPI nelle immagini con Amazon Rekognition utilizzando un SDK AWS
- Rileva le entità nel testo estratto da un'immagine utilizzando un SDK AWS
- Rileva i volti in un'immagine utilizzando un SDK AWS
- Rileva oggetti nelle immagini con Amazon Rekognition utilizzando un SDK AWS
- Rileva persone e oggetti in un video con Amazon Rekognition utilizzando un SDK AWS
- Salva EXIF e altre informazioni sull'immagine utilizzando un SDK AWS
- Trasforma i dati per la tua applicazione con S3 Object Lambda

Creazione di un'app Amazon Transcribe

L'esempio di codice seguente mostra come utilizzare Amazon Transcribe per trascrivere e visualizzare le registrazioni vocali nel browser.

JavaScript

SDK per JavaScript (v3)

Crea un'app che utilizza Amazon Transcribe per trascrivere e visualizzare le registrazioni vocali nel browser. L'app utilizza due bucket Amazon Simple Storage Service (Amazon S3), uno per ospitare il codice dell'applicazione e l'altro per archiviare le trascrizioni. L'app utilizza un pool di utenti Amazon Cognito per autenticare gli utenti. Gli utenti autenticati dispongono delle autorizzazioni AWS Identity and Access Management (IAM) per accedere ai servizi richiesti. AWS

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su. GitHub

Questo esempio è anche disponibile nella Guida per lo sviluppatore di AWS SDK for JavaScript v3 .

Servizi utilizzati in questo esempio

- Amazon Cognito Identity
- Amazon S3
- Amazon Transcribe

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Convertire testo in voce e viceversa utilizzando un AWS SDK

L'esempio di codice seguente mostra come:

- Utilizzare Amazon Polly per sintetizzare un file di input in testo normale (UTF-8) in un file audio.
- Carica il file audio in un bucket Amazon S3.
- Utilizzare Amazon Transcribe per convertire il file audio in testo.
- Visualizzare il testo.

Rust

SDK per Rust

Utilizza Amazon Polly per sintetizzare un file di input di testo normale (UTF-8) in un file audio, caricare il file audio in un bucket Amazon S3, utilizzare Amazon Transcribe per convertire il file audio in testo e visualizzare il testo.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- Amazon Polly

- Amazon S3
- Amazon Transcribe

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Creazione di un'applicazione di gestione delle risorse fotografiche che consente agli utenti di gestire le foto utilizzando etichette

Nell'esempio di codice seguente viene illustrato come creare un'applicazione serverless che consente agli utenti di gestire le foto mediante etichette.

.NET

AWS SDK for .NET

Mostra come sviluppare un'applicazione per la gestione delle risorse fotografiche che rileva le etichette nelle immagini utilizzando Amazon Rekognition e le archivia per recuperarle in seguito.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su GitHub.

Per approfondire l'origine di questo esempio, consulta il post su AWS Community.

Servizi utilizzati in questo esempio

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

C++

SDK per C++

Mostra come sviluppare un'applicazione per la gestione delle risorse fotografiche che rileva le etichette nelle immagini utilizzando Amazon Rekognition e le archivia per recuperarle in seguito.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su GitHub.

Per approfondire l'origine di questo esempio, consulta il post su AWS Community.

Servizi utilizzati in questo esempio

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Java

SDK per Java 2.x

Mostra come sviluppare un'applicazione per la gestione delle risorse fotografiche che rileva le etichette nelle immagini utilizzando Amazon Rekognition e le archivia per recuperarle in seguito.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su GitHub.

Per approfondire l'origine di questo esempio, consulta il post su AWS Community.

Servizi utilizzati in questo esempio

- API Gateway
- DynamoDB
- Lambda

- Amazon Rekognition
- Amazon S3
- Amazon SNS

JavaScript

SDK per JavaScript (v3)

Mostra come sviluppare un'applicazione per la gestione delle risorse fotografiche che rileva le etichette nelle immagini utilizzando Amazon Rekognition e le archivia per recuperarle in seguito.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su GitHub.

Per approfondire l'origine di questo esempio, consulta il post su AWS Community.

Servizi utilizzati in questo esempio

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Kotlin

SDK per Kotlin

Mostra come sviluppare un'applicazione per la gestione delle risorse fotografiche che rileva le etichette nelle immagini utilizzando Amazon Rekognition e le archivia per recuperarle in seguito.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su GitHub.

Per approfondire l'origine di questo esempio, consulta il post su AWS Community.

Servizi utilizzati in questo esempio

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

PHP

SDK per PHP

Mostra come sviluppare un'applicazione per la gestione delle risorse fotografiche che rileva le etichette nelle immagini utilizzando Amazon Rekognition e le archivia per recuperarle in seguito.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su GitHub.

Per approfondire l'origine di questo esempio, consulta il post su AWS Community.

Servizi utilizzati in questo esempio

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Rust

SDK per Rust

Mostra come sviluppare un'applicazione per la gestione delle risorse fotografiche che rileva le etichette nelle immagini utilizzando Amazon Rekognition e le archivia per recuperarle in seguito.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su GitHub.

Per approfondire l'origine di questo esempio, consulta il post su AWS Community.

Servizi utilizzati in questo esempio

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Creazione di un'applicazione Amazon Textract explorer

Gli esempi di codice seguenti mostrano come esplorare l'output di Amazon Textract tramite un'applicazione interattiva.

JavaScript

SDK per JavaScript (v3)

Mostra come utilizzare per AWS SDK for JavaScript creare un'applicazione React che utilizza Amazon Textract per estrarre dati dall'immagine di un documento e visualizzarli in una pagina Web interattiva. Questo esempio viene eseguito in un browser Web e richiede, come credenziali, un'identità autenticata Amazon Cognito. Utilizza Amazon Simple Storage Service (Amazon S3) per l'archiviazione e per le notifiche esegue il polling di una coda di Servizio di coda semplice Amazon (Amazon SQS) sottoscritta a un argomento Servizio di notifica semplice Amazon (Amazon SNS).

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- Amazon Cognito Identity
- Amazon S3
- Amazon SNS
- Amazon SQS
- Amazon Textract

Python

SDK per Python (Boto3)

Mostra come utilizzarlo AWS SDK for Python (Boto3) con Amazon Textract per rilevare elementi di testo, moduli e tabelle nell'immagine di un documento. L'immagine di input e l'output di Amazon Textract sono mostrati in un'applicazione Tkinter che consente di esplorare gli elementi rilevati.

- Invia un'immagine del documento ad Amazon Textract ed esplora l'output degli elementi rilevati.
- Invia immagini direttamente ad Amazon Textract o tramite un bucket Amazon Simple Storage Service (Amazon S3).
- Utilizza le API asincrone per avviare un processo che pubblica una notifica in un argomento Amazon Simple Notification Service (Amazon SNS) al suo termine.
- Esegue il polling di una coda Amazon Simple Queue Service (Amazon SQS) per un messaggio di completamento del processo e visualizza i risultati.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- Amazon S3
- Amazon SNS
- Amazon SQS
- Amazon Textract

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Rileva i DPI nelle immagini con Amazon Rekognition utilizzando un SDK AWS

Gli esempi di codice seguenti mostrano come creare un'applicazione che utilizza Amazon Rekognition per rilevare dispositivi di protezione individuale (DPI) nelle immagini.

Java

SDK per Java 2.x

Mostra come creare una AWS Lambda funzione che rileva le immagini con dispositivi di protezione individuale.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- DynamoDB
- Amazon Rekognition
- Amazon S3
- Amazon SES

JavaScript

SDK per JavaScript (v3)

Mostra come usare Amazon Rekognition AWS SDK for JavaScript con la per creare un'applicazione per rilevare i dispositivi di protezione individuale (DPI) nelle immagini che si trovano in un bucket Amazon Simple Storage Service (Amazon S3). L'applicazione salva i risultati in una tabella Amazon DynamoDB e invia all'amministratore una notifica e-mail sui risultati tramite Amazon Simple Email Service (Amazon SES).

Scopri come:

- Creare un utente non autenticato tramite Amazon Cognito.

- Analizzare le immagini per rilevare i DPI tramite Amazon Rekognition.
- Verificare un indirizzo e-mail per Amazon SES.
- Aggiornare una tabella DynamoDB con i risultati.
- Inviare una notifica e-mail tramite Amazon SES.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su. GitHub

Servizi utilizzati in questo esempio

- DynamoDB
- Amazon Rekognition
- Amazon S3
- Amazon SES

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Rileva le entità nel testo estratto da un'immagine utilizzando un SDK AWS

L'esempio di codice seguente mostra come utilizzare Amazon Comprehend per rilevare le entità nel testo estratto da Amazon Textract da un'immagine archiviata in Amazon S3.

Python

SDK per Python (Boto3)

Mostra come utilizzarlo AWS SDK for Python (Boto3) in un notebook Jupyter per rilevare entità nel testo estratto da un'immagine. In questo esempio viene utilizzato Amazon Textract per estrarre il testo da un'immagine archiviata in Amazon Simple Storage Service (Amazon S3) e Amazon Comprehend per rilevare le entità nel testo estratto.

Questo esempio è un notebook Jupyter e deve essere eseguito in un ambiente in grado di ospitare notebook. Per istruzioni su come eseguire l'esempio utilizzando Amazon SageMaker, consulta le istruzioni in TextractAndComprehendNotebook.ipynb.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su. GitHub

Servizi utilizzati in questo esempio

- Amazon Comprehend
- Amazon S3
- Amazon Textract

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Rileva i volti in un'immagine utilizzando un SDK AWS

L'esempio di codice seguente mostra come:

- Salva un'immagine in un bucket Amazon S3.
- Utilizza Amazon Rekognition per rilevare i dettagli del viso, come fascia di età, sesso ed emozione (ad esempio sorridente).
- Visualizzare questi dettagli.

Rust

SDK per Rust

Salva l'immagine in un bucket Amazon S3 con il prefisso uploads, utilizza Amazon Rekognition per rilevare i dettagli del viso, come fascia di età, sesso ed emozione (sorridente, ecc.) e visualizza tali dettagli.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- Amazon Rekognition
- Amazon S3

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Rileva oggetti nelle immagini con Amazon Rekognition utilizzando un SDK AWS

Gli esempi di codice seguenti mostrano come creare un'applicazione che utilizza Amazon Rekognition per rilevare oggetti in base a una categoria nelle immagini.

.NET

AWS SDK for .NET

Mostra come utilizzare l'API .NET di Amazon Rekognition per creare un'applicazione che utilizza Amazon Rekognition per identificare gli oggetti in base a una categoria nelle immagini situate in un bucket Amazon Simple Storage Service (Amazon S3). L'applicazione invia all'amministratore una notifica e-mail sui risultati tramite Amazon Simple Email Service (Amazon SES).

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- Amazon Rekognition
- Amazon S3
- Amazon SES

Java

SDK per Java 2.x

Mostra come utilizzare l'API Java di Amazon Rekognition per creare un'applicazione che utilizza Amazon Rekognition per identificare gli oggetti in base a una categoria nelle immagini situate in un bucket Amazon Simple Storage Service (Amazon S3). L'applicazione invia all'amministratore una notifica e-mail sui risultati tramite Amazon Simple Email Service (Amazon SES).

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- Amazon Rekognition

- Amazon S3
- Amazon SES

JavaScript

SDK per JavaScript (v3)

Mostra come usare Amazon Rekognition AWS SDK for JavaScript con la per creare un'app che utilizzi Amazon Rekognition per identificare gli oggetti per categoria nelle immagini che si trovano in un bucket Amazon Simple Storage Service (Amazon S3). L'applicazione invia all'amministratore una notifica e-mail sui risultati tramite Amazon Simple Email Service (Amazon SES).

Scopri come:

- Creare un utente non autenticato tramite Amazon Cognito.
- Analizza le immagini per rilevare gli oggetti tramite Amazon Rekognition.
- Verificare un indirizzo e-mail per Amazon SES.
- Inviare una notifica e-mail tramite Amazon SES.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su. GitHub

Servizi utilizzati in questo esempio

- Amazon Rekognition
- Amazon S3
- Amazon SES

Kotlin

SDK per Kotlin

Mostra come utilizzare l'API Kotlin di Amazon Rekognition per creare un'applicazione che utilizza Amazon Rekognition per identificare gli oggetti in base a una categoria nelle immagini situate in un bucket Amazon Simple Storage Service (Amazon S3). L'applicazione invia all'amministratore una notifica e-mail sui risultati tramite Amazon Simple Email Service (Amazon SES).

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- Amazon Rekognition
- Amazon S3
- Amazon SES

Python

SDK per Python (Boto3)

Illustra come utilizzare il AWS SDK for Python (Boto3) per creare un'applicazione Web che consenta di eseguire le seguenti operazioni:

- Caricamento di foto in un bucket Amazon Simple Storage Service (Amazon S3).
- Utilizzo di Amazon Rekognition per analizzare ed etichettare le foto.
- Utilizzo di Amazon Simple Email Service (Amazon SES) per inviare report dell'analisi delle immagini tramite e-mail.

Questo esempio contiene due componenti principali: una pagina web scritta in JavaScript che è costruita con React e un servizio REST scritto in Python creato con Flask-RESTful.

È possibile utilizzare la pagina Web React per:

- Visualizzare un elenco di immagini archiviate nel bucket S3.
- Caricare le immagini dal computer nel bucket S3.
- Visualizzare immagini ed etichette che identificano gli elementi rilevati nell'immagine.
- Ottenere un report relativo a tutte le immagini nel bucket S3 e inviarlo tramite email.

La pagina Web richiama il servizio REST. Il servizio invia richieste a AWS per eseguire le seguenti operazioni:

- Ottenere e filtrare l'elenco delle immagini nel bucket S3.
- Caricare le foto nel bucket S3.
- Utilizzare Amazon Rekognition per analizzare le singole foto e ottenere un elenco di etichette che identificano gli articoli rilevati al loro interno.

- Analizzare tutte le foto presenti nel bucket S3 e usare Amazon SES per inviare un report tramite e-mail.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su. GitHub

Servizi utilizzati in questo esempio

- Amazon Rekognition
- Amazon S3
- Amazon SES

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Rileva persone e oggetti in un video con Amazon Rekognition utilizzando un SDK AWS

Gli esempi di codice seguenti mostrano come rilevare persone e oggetti in un video con Amazon Rekognition.

Java

SDK per Java 2.x

Mostra come utilizzare l'API Java di Amazon Rekognition per creare un'applicazione che rileva volti e oggetti nei video situati in un bucket Amazon Simple Storage Service (Amazon S3). L'applicazione invia all'amministratore una notifica e-mail sui risultati tramite Amazon Simple Email Service (Amazon SES).

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- Amazon Rekognition
- Amazon S3
- Amazon SES

JavaScript

SDK per JavaScript (v3)

Mostra come usare Amazon Rekognition AWS SDK for JavaScript con la per creare un'app per rilevare volti e oggetti nei video che si trovano in un bucket Amazon Simple Storage Service (Amazon S3). L'applicazione invia all'amministratore una notifica e-mail sui risultati tramite Amazon Simple Email Service (Amazon SES).

Scopri come:

- Creare un utente non autenticato tramite Amazon Cognito.
- Analizzare le immagini per rilevare i DPI tramite Amazon Rekognition.
- Verificare un indirizzo e-mail per Amazon SES.
- Inviare una notifica e-mail tramite Amazon SES.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su. GitHub

Servizi utilizzati in questo esempio

- Amazon Rekognition
- Amazon S3
- Amazon SES

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Salva EXIF e altre informazioni sull'immagine utilizzando un SDK AWS

L'esempio di codice seguente mostra come:

- Recuperare informazioni EXIF da un file JPG, JPEG o PNG.
- Carica il file immagine in un bucket Amazon S3.
- Utilizza Amazon Rekognition per identificare i tre attributi principali (etichette) nel file.
- Aggiungi le informazioni su EXIF ed etichette a una tabella Amazon DynamoDB nella regione.

Rust

SDK per Rust

Recupera le informazioni EXIF da un file JPG, JPEG o PNG, carica il file di immagine in un bucket Amazon S3, utilizza Amazon Rekognition per identificare i tre attributi principali (etichette in Amazon Rekognition) nel file e aggiungi le informazioni su EXIF ed etichette a una tabella Amazon DynamoDB nella regione.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- DynamoDB
- Amazon Rekognition
- Amazon S3

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Trasforma i dati per la tua applicazione con S3 Object Lambda

Il seguente esempio di codice mostra come trasformare i dati per la tua applicazione con S3 Object Lambda.

.NET

AWS SDK for .NET

Mostra come aggiungere codice personalizzato alle richieste S3 GET standard per modificare l'oggetto richiesto recuperato da S3 in modo che l'oggetto soddisfi le esigenze del client o dell'applicazione richiedente.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, vedi l'esempio completo su. GitHub

Servizi utilizzati in questo esempio

- Lambda

- Amazon S3

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Risoluzione dei problemi

In questa sezione viene descritto come risolvere i problemi relativi alle funzionalità di Amazon S3 e come ottenere gli ID richiesta necessari per contattare AWS Support.

Argomenti

Risoluzione dei problemi relativi agli errori di accesso negato (403 Accesso negato) in Amazon S3

> **Important**
>
> Il 13 maggio 2024, abbiamo iniziato a implementare una modifica per eliminare gli addebiti per le richieste non autorizzate che non sono state avviate dal proprietario del bucket. Una volta completata l'implementazione di questa modifica, i proprietari dei bucket non dovranno mai sostenere costi di richiesta o larghezza di banda per le richieste che restituiscono errori `AccessDenied` (HTTP`403 Forbidden`) quando tali richieste vengono avviate dall'esterno del loro account o organizzazione individuale. AWS AWS Per ulteriori informazioni sull'elenco completo dei codici HTTP `3XX` e di `4XX` stato che non verranno fatturati, consulta. Risposte agli errori di fatturazione per Amazon S3 Questa modifica alla fatturazione non richiede aggiornamenti alle applicazioni e si applica a tutti i bucket S3. Una volta completata l'implementazione di questa modifica Regioni AWS, aggiorneremo la nostra documentazione.

I seguenti argomenti trattano le cause più comuni degli errori di accesso negato (403 Accesso negato) in Amazon S3.

> **Note**
>
> Per `Access Denied` (HTTP`403 Forbidden`), S3 non addebita alcun costo al proprietario del bucket quando la richiesta viene avviata al di fuori dell' AWS account individuale del proprietario del bucket o dell'organizzazione del proprietario del bucket. AWS

Argomenti

> **Note**
>
> Se stai cercando di risolvere un problema di autorizzazioni, inizia con la sezione Policy del bucket e policy IAM e assicurati di seguire le indicazioni contenute in Suggerimenti per il controllo delle autorizzazioni.

Policy di bucket e policy IAM

Operazioni a livello di bucket

Se non esiste una policy relativa al bucket, il bucket consente implicitamente le richieste provenienti da qualsiasi identità AWS Identity and Access Management (IAM) nell'account proprietario del bucket. Inoltre, il bucket rifiuta implicitamente le richieste provenienti da qualsiasi altra identità IAM da qualsiasi altro account e le richieste anonime (non firmate). Tuttavia, se non è implementata alcuna policy utente IAM, al richiedente (a meno che non sia l'utente root) viene implicitamente negato di effettuare richieste. Per ulteriori informazioni su questa logica di valutazione, consulta Determinazione se una richiesta è consentita o rifiutata in un account nella Guida per l'utente di IAM.

Operazioni a livello di oggetti

Se l'oggetto è di proprietà dell'account proprietario del bucket, la policy di bucket e la policy degli utenti IAM funzioneranno allo stesso modo per le operazioni a livello di oggetto e per le operazioni a livello di bucket. Ad esempio, se non è stata implementata alcuna policy di bucket, il bucket consente implicitamente di eseguire le richieste provenienti da qualsiasi identità IAM nell'account proprietario del bucket. Inoltre, il bucket rifiuta implicitamente le richieste di oggetti provenienti da qualsiasi altra identità IAM da qualsiasi altro account e le richieste anonime (non firmate). Tuttavia, se non è implementata alcuna policy utente IAM, al richiedente (a meno che non sia l'utente root) viene implicitamente negato di effettuare richieste di oggetti.

Se l'oggetto è di proprietà di un account esterno, l'accesso all'oggetto può essere concesso solo tramite le relative liste di controllo degli accessi (ACL). La policy di bucket e la policy degli utenti IAM possono ancora essere utilizzate per rifiutare le richieste di oggetti.

Pertanto, per assicurarti che la policy di bucket o la policy degli utenti IAM non restituiscano un errore di accesso negato (403 Accesso negato), assicurati che siano soddisfatti i seguenti requisiti:

- Per l'accesso con lo stesso account, non deve esserci un'istruzione `Deny` esplicita per il richiedente a cui stai cercando di concedere le autorizzazioni nella policy di bucket né nella politica degli utenti IAM. Se desideri concedere le autorizzazioni utilizzando solo la policy di bucket e la policy degli utenti IAM, deve esserci almeno un'istruzione `Allow` esplicita in una di queste policy.
- Per l'accesso multi-account, non deve esserci un'istruzione `Deny` esplicita per il richiedente a cui stai cercando di concedere le autorizzazioni nella policy di bucket né nella politica degli utenti IAM. Se desideri concedere autorizzazioni multi-account utilizzando solo la policy di bucket e la policy degli utenti IAM, sia la policy di bucket che la policy degli utenti IAM del richiedente devono includere un'istruzione `Allow` esplicita.

> **Note**
>
> Le istruzioni `Allow` in una policy di bucket si applicano solo agli oggetti di proprietà dello stesso account proprietario del bucket. Tuttavia, le istruzioni `Deny` in una policy di bucket si applicano a tutti gli oggetti indipendentemente dalla proprietà dell'oggetto.

Per rivedere o modificare la policy di bucket

> **Note**
>
> Per visualizzare o modificare una policy di bucket, devi disporre dell'autorizzazione `s3:GetBucketPolicy`.

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket scegli il nome del bucket per il quale vuoi visualizzare o modificare una policy di bucket.
4. Scegli la scheda Autorizzazioni.
5. In Bucket Policy (Policy del bucket) scegliere Edit (Modifica). Viene visualizzata la pagina Edit bucket policy (Modifica policy di bucket).

Per rivedere o modificare la tua policy sui bucket utilizzando AWS Command Line Interface (AWS CLI), usa il get-bucket-policycomando.

> **Note**
>
> Se rimani bloccato all'accesso a un bucket a causa di una policy di bucket errata, accedi AWS Management Console utilizzando le credenziali dell'utente root. Per riottenere l'accesso al bucket, assicurati di eliminare la policy di bucket utilizzando le tue credenziali di utente root.

Suggerimenti per la verifica delle autorizzazioni

Per verificare se il richiedente dispone delle autorizzazioni adeguate per eseguire un'operazione Amazon S3, prova quanto segue:

- Identifica il richiedente. Se si tratta di una richiesta non firmata, significa che è una richiesta anonima senza una policy utente IAM. Se si tratta di una richiesta che utilizza un URL prefirmato, la policy utente sarà la stessa del ruolo o utente IAM che ha firmato la richiesta.

- Assicurati di utilizzare il ruolo o l'utente IAM corretto. Per verificare il ruolo o utente IAM, controlla nell'angolo in alto a destra della AWS Management Console o utilizza il comando aws sts get-caller-identity.
- Controlla tutte le policy IAM collegate al ruolo o all'utente IAM. È possibile utilizzare uno dei seguenti metodi:
  - Verifica le policy IAM con il simulatore di policy IAM.
  - Esamina i vari tipi di policy IAM.
- Se necessario, modifica la policy utente IAM.
- Consulta i seguenti esempi di policy che negano o consentono esplicitamente l'accesso:
  - Policy utente IAM di autorizzazione esplicita: IAM: consente e rifiuta l'accesso a più servizi a livello di programmazione e nella console
  - Policy di bucket di autorizzazione esplicita: Concessione delle autorizzazioni a più account per caricare oggetti o impostare le ACL degli oggetti per l'accesso pubblico
  - Politica esplicita di negazione degli utenti IAM AWS: Nega l'accesso a in base alla richiesta AWS Regione AWS
  - Policy di bucket di rifiuto esplicito: Richiesta SSE-KMS per tutti gli oggetti scritti in un bucket

Impostazioni ACL di Amazon S3

Quando controlli le impostazioni relative alle le liste di controllo degli accessi, controlla innanzitutto l'impostazione dell'opzione Proprietà dell'oggetto per verificare se le liste di controllo degli accessi sono abilitate nel bucket. Tieni presente che le autorizzazioni ACL possono essere utilizzate solo per concedere autorizzazioni e non per rifiutare le richieste. Inoltre, le liste di controllo degli accessi (ACL) non possono essere utilizzate per concedere l'accesso ai richiedenti rifiutati esplicitamente nelle policy di bucket o nelle policy degli utenti IAM.

L'opzione Proprietà dell'oggetto è impostata su Bucket owner enforced.

Se è abilitata l'impostazione Bucket owner enforced, è improbabile che le impostazioni ACL causino un errore di accesso negato (403 Accesso negato) perché questa impostazione disabilita tutte le ACL valide per il bucket e gli oggetti. Bucket owner enforced è l'impostazione predefinita (e consigliata) per i bucket Amazon S3.

L'opzione Proprietà dell’oggetto è impostata su Proprietario del bucket preferito o Autore dell'oggetto

Le autorizzazioni ACL continuano a essere valide con l'impostazione Proprietario del bucket preferito o Autore dell'oggetto. Esistono due tipi di ACL: le ACL dei bucket e le ACL degli oggetti. Per informazioni sulle differenze tra questi due tipi di ACL, consulta Mappatura delle autorizzazioni ACL e delle autorizzazioni della policy di accesso.

A seconda dell'azione della richiesta rifiutata, controlla le autorizzazioni ACL per il bucket o l'oggetto:

- Se Amazon S3 ha rifiutato una richiesta `LIST`, `PUT` oggetto, `GetBucketAcl` o `PutBucketAcl`, controlla le autorizzazioni ACL per il tuo bucket.

  > **Note**
  >
  > Non è possibile concedere autorizzazioni oggetto `GET` con le impostazioni ACL del bucket.

- Se Amazon S3 ha rifiutato una richiesta `GET` per un oggetto S3 o una richiesta PutObjectAcl, controlla le autorizzazioni ACL per l'oggetto.

  > **Important**
  >
  > Se l'account proprietario dell'oggetto è diverso dall'account proprietario del bucket, l'accesso all'oggetto non è controllato dalla policy di bucket.

Risoluzione di un errore di accesso negato (403 Accesso negato) derivante da una richiesta oggetto **`GET`** durante la proprietà di un oggetto multi-account

Esamina le impostazioni dell'opzione Proprietà dell’oggetto del bucket per determinare il proprietario dell'oggetto. Se hai accesso alle ACL dell'oggetto, puoi anche controllare l'account del proprietario dell'oggetto. (Per visualizzare l'account del proprietario dell'oggetto, controlla l'impostazione ACL dell'oggetto nella console Amazon S3.) In alternativa, puoi anche eseguire una richiesta `GetObjectAcl` per trovare l'ID canonico del proprietario dell'oggetto per verificare l'account del proprietario. Per impostazione predefinita, le ACL concedono autorizzazioni di tipo autorizzazione esplicita per le richieste `GET` all'account del proprietario dell'oggetto.

Dopo aver verificato che il proprietario dell'oggetto sia diverso dal proprietario del bucket, a seconda del caso d'uso e del livello di accesso, scegli uno dei seguenti metodi per risolvere l'errore di accesso negato (403 Accesso negato):

- Disabilitare le ACL (consigliato): questo metodo è valido per tutti gli oggetti e può essere eseguito dal proprietario del bucket. Questo metodo assegna automaticamente il ruolo di proprietario del bucket e il controllo completo su ogni oggetto in esso contenuto. Prima di implementare questo metodo, verifica i prerequisiti per la disabilitazione delle ACL. Per informazioni su come impostare il bucket su Bucket owner enforced (consigliata), consulta l'argomento relativo all'impostazione della proprietà dell'oggetto su un bucket esistente.

  > **Important**
  >
  > Per evitare un errore di accesso negato (403 Accesso negato), assicurati di eseguire la migrazione delle autorizzazioni delle liste di controllo degli accessi a una policy di bucket prima di disabilitare le liste. Per ulteriori informazioni, consulta Esempi di policy di bucket per la migrazione delle autorizzazioni delle ACL.

- Cambiare il proprietario dell'oggetto in proprietario del bucket: questo metodo può essere applicato a singoli oggetti, ma solo il proprietario dell'oggetto (o un utente con le autorizzazioni appropriate) può modificare la proprietà di un oggetto. Potrebbero venire applicati costi PUT aggiuntivi. Per ulteriori informazioni, consulta la pagina Prezzi di Amazon S3. Questo metodo garantisce al proprietario del bucket la piena proprietà dell'oggetto, consentendogli di controllare l'accesso all'oggetto tramite una policy di bucket.

  Per modificare la proprietà dell'oggetto, procedi in uno dei seguenti modi:

  - Tu (il proprietario del bucket) puoi copiare nuovamente l'oggetto nel bucket.
  - È possibile modificare l'impostazione dell'opzione Proprietà dell’oggetto per il bucket impostandola su Proprietario del bucket preferito. Se il controllo delle versioni è disabilitato, gli oggetti nel bucket vengono sovrascritti. Se il controllo delle versioni è abilitato, nel bucket verranno visualizzate versioni duplicate dello stesso oggetto, per le quali il proprietario può impostare una regola del ciclo di vita per la scadenza. Per istruzioni su come modificare le impostazioni dell'opzione Proprietà dell’oggetto, consulta Impostazione di Object Ownership su un bucket esistente.

> **Note**
>
> Quando aggiorni l'impostazione dell'opzione Proprietà dell'oggetto impostandola su Proprietario del bucket preferito, l'impostazione viene applicata solo ai nuovi oggetti caricati nel bucket.

- È possibile fare in modo che il proprietario dell'oggetto carichi nuovamente l'oggetto con l'ACL `bucket-owner-full-control` predefinita dell'oggetto.

> **Note**
>
> Per i caricamenti multi-account, nella tua policy di bucket dovrai disporre anche dell'ACL `bucket-owner-full-control` dell'oggetto predefinito. Per una policy di bucket di esempio, consulta Concedere autorizzazioni multi-account per il caricamento di oggetti a garanzia del controllo completo da parte del proprietario del bucket.

- Mantenere l'autore dell'oggetto come proprietario dell'oggetto: questo metodo non modifica il proprietario dell'oggetto, ma consente di concedere l'accesso agli oggetti singolarmente. Per concedere l'accesso a un oggetto, devi disporre dell'autorizzazione `PutObjectAcl` per l'oggetto. Quindi, per correggere l'errore di accesso negato (403 Accesso negato), aggiungi il richiedente come utente autorizzato per accedere all'oggetto nelle relative ACL. Per ulteriori informazioni, consulta Configurazione delle ACL.

Impostazioni dell'opzione S3 Blocco dell'accesso pubblico

Se la richiesta non riuscita riguarda l'accesso pubblico o le policy pubbliche, controlla le impostazioni S3 dell'opzione Blocco dell'accesso pubblico sul tuo account, bucket o punto di accesso S3. A partire da aprile 2023, tutte le impostazioni dell'opzione Blocco dell'accesso pubblico sono abilitate per impostazione predefinita per i nuovi bucket. Per ulteriori informazioni su cosa si intende con il termine "pubblico" in Amazon S3, consulta Significato di "pubblico".

Se impostate su `TRUE`, le impostazioni dell'opzione Blocco dell'accesso fungono da policy di rifiuto esplicito che sostituiscono le autorizzazioni consentite dalle ACL, dalle policy di bucket e dalle policy degli utenti IAM. Per determinare se le impostazioni dell'opzione Blocco dell'accesso stanno rifiutando la tua richiesta, esamina i seguenti scenari:

- Se la lista di controllo degli accessi specificata è pubblica, l'impostazione `BlockPublicAcls`comporta il rifiuto delle chiamate `PutBucketAcl` e `PutObjectACL`.
- Se la richiesta include una ACL pubblica, l'impostazione `BlockPublicAcls` comporta il rifiuto delle chiamate `PutObject`.
- Se l'impostazione `BlockPublicAcls` viene applicata a un account e la richiesta include una ACL pubblica, tutte le chiamate `CreateBucket` che includono ACL pubbliche hanno esito negativo.
- Se l'autorizzazione della richiesta è concessa solo da una ACL pubblica, l'impostazione `IgnorePublicAcls` comporta il rifiuto della richiesta.
- Se la policy di bucket specificata consente l'accesso pubblico, l'impostazione `BlockPublicPolicy` rifiuta le chiamate `PutBucketPolicy`.
- Se l'impostazione `BlockPublicPolicy` viene applicata a un punto di accesso, tutte le chiamate `PutAccessPointPolicy` e `PutBucketPolicy` che specificano una policy pubblica ed effettuate tramite il punto di accesso avranno esito negativo.
- Se il punto di accesso o il bucket ha una politica pubblica, l'`RestrictPublicBuckets`impostazione rifiuta tutte le chiamate tra account tranne quelle principali. Servizio AWS Questa impostazione rifiuta anche tutte le chiamate anonime (o non firmate).

Per rivedere e aggiornare le configurazioni delle impostazioni dell'opzione Blocco dell'accesso pubblico, consulta Configurazione delle impostazioni di blocco dell'accesso pubblico per i bucket S3.

Impostazioni della crittografia Amazon S3

Amazon S3 supporta la crittografia lato server nel bucket. La crittografia lato server è la crittografia dei dati nella posizione di destinazione eseguita dall'applicazione o dal servizio che li riceve. Amazon S3 crittografa i dati a livello di oggetto mentre li scrive su dischi nei data AWS center e li decrittografa per te quando vi accedi.

Per impostazione predefinita, Amazon S3 ora applica la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3) come livello di base della crittografia per ogni bucket di Amazon S3. Amazon S3 consente inoltre di specificare il metodo di crittografia lato server durante il caricamento degli oggetti.

Per esaminare lo stato della crittografia lato server e le impostazioni della crittografia del bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.

2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket, scegli il bucket per cui vuoi controllare le impostazioni della crittografia.
4. Scegliere la scheda Properties (Proprietà).
5. Scorri verso il basso fino alla sezione Crittografia predefinita e visualizza le impostazioni dell'opzione Tipo di crittografia.

Per verificare le impostazioni di crittografia utilizzando il AWS CLI, usa il get-bucket-encryptioncomando.

Per controllare lo stato della crittografia dell'oggetto

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket scegli il nome del bucket contenente l'oggetto.
4. Nell'elenco Nome scegli il nome dell'oggetto per cui desideri aggiungere o modificare la crittografia.

   Viene visualizzata la pagina dei dettagli dell'oggetto.
5. Scorri verso il basso fino alla sezione Impostazioni crittografia lato server per visualizzare le impostazioni della crittografia lato server dell'oggetto.

Per verificare lo stato di crittografia degli oggetti utilizzando il AWS CLI, usa il head-objectcomando.

Requisiti relativi a crittografia e autorizzazioni

Amazon S3 supporta tre tipi di crittografia lato server:

- Crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3)
- Crittografia lato server con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS)
- Crittografia lato server con chiavi fornite dal cliente (SSE-C)

In base alle impostazioni di crittografia correnti, verifica che siano soddisfatti i seguenti requisiti relativi alle autorizzazioni:

- SSE-S3: non sono richieste autorizzazioni aggiuntive.

- SSE-KMS (con una chiave gestita dal cliente): per caricare oggetti, è necessaria l'autorizzazione `kms:GenerateDataKey` per la AWS KMS key . Per scaricare oggetti ed eseguire caricamenti di oggetti in più parti, è necessaria l'autorizzazione `kms:Decrypt` per la chiave KMS.
- SSE-KMS (con un Chiave gestita da AWS): il richiedente deve appartenere allo stesso account proprietario della chiave KMS. `aws/s3` Il richiedente deve inoltre disporre delle autorizzazioni Amazon S3 corrette per accedere all'oggetto.
- SSE-C (con una chiave fornita dal cliente): non sono richieste autorizzazioni aggiuntive. Puoi configurare la policy di bucket per richiedere e limitare la crittografia lato server con chiavi di crittografia fornite dal cliente per gli oggetti nel bucket.

Se l'oggetto è crittografato con una chiave gestita dal cliente, assicurati che la policy della chiave KMS ti consenta di eseguire le operazioni `kms:GenerateDataKey` o `kms:Decrypt`. Per istruzioni su come verificare la policy della chiave KMS, consulta Visualizzazione di una policy di chiave nella Guida per gli sviluppatori di AWS Key Management Service .

Impostazioni dell'opzione S3 Blocco oggetti

Se nel bucket è abilitata la funzionalità S3 Blocco oggetti e l'oggetto è protetto da un periodo di conservazione o da un blocco a fini legali, Amazon S3 restituisce un errore di accesso negato (403 Accesso negato) quando si tenta di eliminare l'oggetto.

Per verificare se l'opzione Blocco oggetti è abilitata per il bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Bucket scegli il nome del bucket da controllare.
4. Scegliere la scheda Properties (Proprietà).
5. Scorri verso il basso fino alla sezione Blocco oggetti. Verifica se l'impostazione dell'opzione Blocco oggetti è abilitata o disabilitata.

Per determinare se l'oggetto è protetto da un periodo di conservazione o da un blocco a fini legali, visualizza le informazioni relative al blocco dell'oggetto.

Se l'oggetto è protetto da un periodo di conservazione o da un blocco a fini legali, verifica quanto segue:

- Se la versione dell'oggetto è protetta dalla modalità di conservazione della conformità, non è possibile eliminarla definitivamente. Una richiesta `DELETE` permanente da parte di qualsiasi richiedente, incluso l'utente root, genererà un errore di accesso negato (403 Accesso negato). Inoltre, considera che quando invii una richiesta `DELETE` per un oggetto protetto dalla modalità di conservazione della conformità, Amazon S3 crea un contrassegno di eliminazione per l'oggetto.
- Se la versione dell'oggetto è protetta con la modalità di conservazione della governance e disponi dell'autorizzazione `s3:BypassGovernanceRetention`, è possibile aggirare la protezione ed eliminare definitivamente la versione. Per ulteriori informazioni, consulta Bypassare la modalità Governance.
- Se la versione dell'oggetto è protetta da un blocco a fini legali, una richiesta `DELETE` permanente può generare un errore di accesso negato (403 Accesso negato). Per eliminare definitivamente la versione dell'oggetto, è necessario rimuovere il blocco a fini legali applicato alla versione dell'oggetto. Per rimuovere un blocco a fini legali, devi disporre dell'autorizzazione `s3:PutObjectLegalHold`. Per ulteriori informazioni sulla rimozione di un blocco a fini legali, consulta Configurazione di S3 Object Lock.

Policy degli endpoint VPC

Se accedi ad Amazon S3 utilizzando un endpoint di cloud privato virtuale (VPC), assicurati che la policy di endpoint VPC non ti impedisca di accedere alle risorse Amazon S3. Per impostazione predefinita, la policy degli endpoint VPC consente di eseguire tutte le richieste indirizzate ad Amazon S3. È inoltre possibile configurare la policy degli endpoint VPC per limitare determinate richieste. Per informazioni su come controllare la policy degli endpoint VPC, consulta Controllo dell'accesso agli endpoint VPC tramite le policy di endpoint nella Guida di AWS PrivateLink .

AWS Organizations politiche

Se fai Account AWS parte di un'organizzazione, AWS Organizations le policy possono impedirti di accedere alle risorse Amazon S3. Per impostazione predefinita, AWS Organizations le policy non bloccano alcuna richiesta ad Amazon S3. Tuttavia, assicurati che AWS Organizations le tue politiche non siano state configurate per bloccare l'accesso ai bucket S3. Per istruzioni su come controllare le tue AWS Organizations politiche, consulta Elenco di tutte le politiche nella Guida per l'AWS Organizations utente.

Impostazioni del punto di accesso

Se ricevi un errore di accesso negato (403 Accesso negato) mentre effettui richieste tramite i punti di accesso Amazon S3, potresti dover controllare quanto segue:

- Le configurazioni per i punti di accesso
- La policy utente IAM utilizzata per i punti di accesso
- La policy di bucket utilizzata per gestire o configurare i punti di accesso multi-account

Configurazioni e policy dei punti di accesso

- Quando si crea un punto di accesso, è possibile scegliere di impostare Internet o VPC come origine della rete. Se l'origine della rete è impostata si "Solo VPC", Amazon S3 rifiuterà tutte le richieste effettuate al punto di accesso che non provengono dal VPC specificato. Per verificare l'origine della rete del punto di accesso, consulta Creazione di access point limitati a un cloud privato virtuale.
- Con i punti di accesso, puoi anche configurare impostazioni personalizzate dell'opzione Blocco dell'accesso pubblico, che funzionano in modo simile alle impostazioni di Blocco dell'accesso pubblico a livello di bucket o account. Per verificare le impostazioni personalizzate dell'opzione Blocco dell'accesso pubblico, consulta Gestione dell'accesso pubblico agli access point.
- Per effettuare con successo richieste ad Amazon S3 utilizzando i punti di accesso, assicurati che il richiedente disponga delle autorizzazioni IAM necessarie. Per ulteriori informazioni, consulta Configurazione delle policy IAM per l'utilizzo degli access point.
- Se la richiesta interessa punti di accesso multi-account, assicurati che il proprietario del bucket abbia aggiornato la policy di bucket per autorizzare le richieste provenienti dal punto di accesso. Per ulteriori informazioni, consulta Concessione delle autorizzazioni per i punti di accesso multi-account.

Se l'errore Accesso negato (403 Forbidden) persiste dopo aver controllato tutti gli elementi di questo argomento, recupera l'ID della richiesta Amazon S3 e contattalo per ulteriori informazioni. AWS Support

Risoluzione dei problemi relativi alle operazioni in batch

Negli argomenti seguenti sono descritti gli errori comuni per capire come risolvere i problemi riscontrati durante le operazioni in batch.

Errori comuni

- Il report del processo non viene distribuito quando esiste un problema di autorizzazioni o è abilitata la modalità di conservazione del blocco degli oggetti S3
- Problemi di replica batch S3 con errore: la generazione del manifesto non ha trovato chiavi corrispondenti ai criteri di filtro
- Gli errori relativi alle operazioni in batch si verificano dopo l'aggiunta di una nuova regola di replica a una configurazione della replica esistente
- Operazioni Batch che non funzionano correttamente con l'errore 400 InvalidRequest: operazione non riuscita a causa della mancanza VersionId
- Errori di creazione di processi con l'opzione Tag dell'attività abilitata
- Accesso negato durante la lettura del manifesto

Il report del processo non viene distribuito quando esiste un problema di autorizzazioni o è abilitata la modalità di conservazione del blocco degli oggetti S3

L'errore seguente si verifica se le autorizzazioni richieste mancano o la modalità di conservazione del blocco degli oggetti (modalità governance o modalità conformità) è abilitata nel bucket di destinazione.

Errore: Motivi dell'errore. Non è stato possibile scrivere il report del processo nel bucket dei report. Controlla le autorizzazioni.

Il ruolo IAM e la policy di attendibilità devono essere configurati per consentire a S3 Batch Operations l'accesso agli oggetti PUT nel bucket in cui verrà distribuito il report. Se queste autorizzazioni obbligatorie mancano, si verifica un errore di distribuzione del report del processo.

Quando è abilitata una modalità di conservazione, il bucket è protetto write-once-read-many (WORM). Blocco oggetti con modalità di conservazione abilitata nel bucket di destinazione non è supportato, pertanto i tentativi di distribuzione del report di completamento del processo non

vanno a buon fine. Per risolvere questo problema, scegli un bucket di destinazione per i report di completamento dei processi in cui non sia abilitata la modalità di conservazione del blocco oggetti.

Problemi di replica batch S3 con errore: la generazione del manifesto non ha trovato chiavi corrispondenti ai criteri di filtro

Errore: La generazione del manifesto non ha trovato chiavi corrispondenti ai criteri di filtro.

Questo errore si verifica per uno dei seguenti motivi:

- Quando gli oggetti nel bucket di origine sono archiviati nelle classi di storage S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive.

  Per utilizzare la replica in batch su questi oggetti, è innanzitutto necessario ripristinarli nella classe di archiviazione S3 standard utilizzando un'operazione S3 Avvia ripristino oggetto in un processo Operazioni in batch. Per ulteriori informazioni, consulta Ripristino di un oggetto archiviato e Ripristino di oggetti (operazioni in batch). Dopo aver ripristinato gli oggetti, è possibile replicarli utilizzando un processo di replica in batch.
- Quando i criteri di filtro forniti non corrispondono a nessun oggetto valido nel bucket di origine.

  Verifica e correggi i criteri di filtro. Ad esempio, nella regola Batch Replication, i criteri di filtro cercano tutti gli oggetti nel bucket *example-s3-con* il prefisso. `Tax/` Se il nome del prefisso è stato inserito in modo errato, con una barra all'inizio e alla fine `/Tax/` anziché solo alla fine, non è stato trovato alcun oggetto S3. Per risolvere l'errore, correggi il prefisso, in questo caso, da `/Tax/` a `Tax/` nella regola di replica.

Gli errori relativi alle operazioni in batch si verificano dopo l'aggiunta di una nuova regola di replica a una configurazione della replica esistente

Il processo Operazioni in batch tenta di eseguire la replica degli oggetti esistenti per ogni regola nella configurazione della replica del bucket di origine. In caso di problemi con una delle regole di replica esistenti, è possibile che vengano restituiti errori.

Il report di completamento del processo Operazioni in batch spiega i motivi della mancata esecuzione del processo. Per visualizzare un elenco di errori comuni, consulta Motivi degli errori di replica Amazon S3.

Operazioni Batch che non funzionano correttamente con l'errore 400 InvalidRequest: operazione non riuscita a causa della mancanza VersionId

L'errore di esempio seguente si verifica se un processo Operazioni in batch esegue operazioni su oggetti in un bucket con il controllo delle versioni abilitato e rileva un oggetto nel manifesto con un campo ID versione vuoto.

Errore: *`BUCKET_NAME, prefix/file_name, failed,400,,`* Attività non riuscita a causa della mancanza InvalidRequest VersionId

Questo errore si verifica perché il campo ID versione nel manifesto è una stringa vuota anziché una stringa `null` letterale.

Le operazioni in batch avranno esito negativo per l'oggetto o gli oggetti specifici, ma non per l'intero processo. Questo problema si verifica se il formato del manifesto è configurato per utilizzare gli ID di versione durante l'operazione. I processi senza controllo delle versioni non restituiscono questo problema perché funzionano solo sulla versione più recente di ciascun oggetto e ignorano gli ID di versione nel manifesto.

Per risolvere questo problema, converti gli ID di versione vuoti in stringhe `null`. Per ulteriori informazioni, consulta the section called “Convertire stringhe di ID versione vuote in stringhe nulle”.

Errori di creazione di processi con l'opzione Tag dell'attività abilitata

Senza l'autorizzazione `s3:PutJobTagging`, la creazione di processi Operazioni in batch con l'opzione Tag dell'attività abilitata causa errori `403 access denied`.

Per creare lavori Batch Operations con l'opzione job tag abilitata, l'utente AWS Identity and Access Management (IAM) che sta creando il processo Batch Operations deve disporre dell'`s3:PutJobTagging`autorizzazione oltre all'`s3:CreateJob`autorizzazione.

Per ulteriori informazioni sulle autorizzazioni necessarie per le operazioni in batch, consulta the section called “Concessione di autorizzazioni”.

Accesso negato durante la lettura del manifesto

Se il processo Operazioni in batch non è in grado di leggere il file del manifesto quando tenti di creare il processo, possono verificarsi i seguenti errori.

AWS CLI

Motivo dell'errore La lettura del manifesto è vietata: AccessDenied

Console Amazon S3

Avviso: impossibile ottenere l'ETag dell'oggetto manifesto. Specifica un oggetto diverso per continuare.

Per risolvere questo problema, esegui le seguenti operazioni:

- Verifica che il ruolo IAM utilizzato per creare il Account AWS job Batch Operations disponga delle `s3:GetObject` autorizzazioni. Il ruolo IAM dell'account deve disporre delle autorizzazioni `s3:GetObject` per consentire al processo Operazioni in batch di leggere il file manifesto.

  Per ulteriori informazioni sulle autorizzazioni necessarie per le operazioni in batch, consulta the section called “Concessione di autorizzazioni”.
- Controlla i metadati degli oggetti manifesto per eventuali discrepanze di accesso con l'opzione S3 Proprietà dell’oggetto. Per ulteriori informazioni sull'opzione S3 Proprietà dell’oggetto, consulta the section called “Controllo della proprietà degli oggetti”.
- Controlla se le chiavi AWS Key Management Service (AWS KMS) vengono utilizzate per crittografare il file manifest.

  Batch Operations supporta report di inventario CSV AWS KMS crittografati. Tuttavia, Batch Operations non supporta i file manifest CSV AWS KMS crittografati. Per ulteriori informazioni, consultare Configurazione di Amazon S3 Inventory e Specifica di un manifest.

Risoluzione dei problemi di CORS

Se si riscontra un comportamento imprevisto quando si accede ai bucket impostati con la configurazione CORS, prova le operazioni di risoluzione dei problemi seguenti:

1. Verificare che la configurazione CORS sia impostata nel bucket.

   Se la configurazione CORS è impostata, nella console è visualizzato un collegamento Edit CORS Configuration (Modifica configurazione CORS) nella sezione Permissions (Autorizzazioni) delle Properties (Proprietà) del bucket.
2. Acquisire la richiesta e la risposta complete utilizzando lo strumento che si desidera. Per ogni richiesta ricevuta da Amazon S3, deve esistere una regola CORS corrispondente ai dati nella richiesta, come indicato di seguito:

a. Verificare che l'intestazione della richiesta sia Origin.

   Se l'intestazione manca, Amazon S3 non considera la richiesta come una richiesta multiorigine e non invia le intestazioni della risposta CORS nella risposta.

b. Verificare che l'intestazione Origin nella richiesta corrisponda ad almeno uno degli elementi `AllowedOrigin` nella `CORSRule` specificata.

   Lo schema, l'host e i valori della porta nell'intestazione della richiesta Origin devono corrispondere agli elementi `AllowedOrigin` nella `CORSRule`. Se ad esempio si imposta la `CORSRule` per consentire l'origine `http://www.example.com`, nessuna delle due origini (`https://www.example.com` e `http://www.example.com:80`) nella richiesta corrisponde all'origine consentita nella configurazione.

c. Verifica che il metodo nella richiesta (o il metodo specificato in `Access-Control-Request-Method` nel caso di una richiesta preliminare) corrisponda a uno degli elementi `AllowedMethod` nella stessa `CORSRule`.

d. Per una richiesta preliminare, se la richiesta include un'intestazione `Access-Control-Request-Headers`, verificare che la `CORSRule` includa le voci `AllowedHeader` per ogni valore nell'intestazione `Access-Control-Request-Headers`.

Risoluzione dei problemi del ciclo di vita di Amazon S3

Le informazioni seguenti possono essere utili per risolvere i problemi con le regole del ciclo di vita di Amazon S3.

Argomenti

- Posso recuperare oggetti S3 scaduti in base alle regole del ciclo di vita di S3?

Ho eseguito un'operazione di elenco sul mio bucket e sono stati visualizzati oggetti che pensavo scaduti o sottoposti a transizione in base a una regola del ciclo di vita.

Le transizioni e le scadenze di oggetti del ciclo di vita S3 sono operazioni asincrone. Pertanto, potrebbe essersi verificato un ritardo tra il momento in cui gli oggetti sono idonei per la scadenza o la transizione e il momento in cui si è effettivamente verificata la transizione o la scadenza. Le modifiche a livello di fatturazione vengono applicate non appena la regola del ciclo di vita viene soddisfatta, anche se l'operazione non è completa. Un'eccezione a questo comportamento è se si dispone di una regola del ciclo di vita per il trasferimento alla classe di archiviazione Piano intelligente Amazon S3. In questo caso, le modifiche alla fatturazione non si verificano fino a quando l'oggetto non è stato trasferito alla classe Piano intelligente Amazon S3. Per ulteriori informazioni sulle modifiche alla fatturazione, consulta Impostazione della configurazione del ciclo di vita in un bucket.

> **Note**
>
> Amazon S3 non esegue una transizione di oggetti di dimensioni inferiori a 128 KB dalla classe di archiviazione Amazon S3 Standard o Accesso Infrequente Amazon S3 Standard (AI S3 Standard) alla classe di archiviazione Piano intelligente Amazon S3, Accesso Infrequente Amazon S3 Standard (AI S3 Standard) o Accesso infrequente a zona unica Amazon S3 (AI a zona unica S3).

Come posso monitorare le azioni intraprese dalle mie regole del ciclo di vita?

Per monitorare le azioni intraprese dalle regole del ciclo di vita, puoi utilizzare le seguenti funzionalità:

- Notifiche degli eventi S3: puoi configurare le notifiche degli eventi S3 in modo da ricevere una notifica di eventuali eventi di scadenza o transizione del ciclo di vita di S3.
- Registri di accesso al server S3: puoi abilitare i log di accesso al server per i tuoi bucket S3 per registrare le azioni del ciclo di vita di S3, come le transizioni degli oggetti verso un'altra classe di storage o le scadenze degli oggetti. Per ulteriori informazioni, consulta Ciclo di vita e registrazione.

Per visualizzare le modifiche nello storage causate dalle azioni del ciclo di vita su base giornaliera, ti consigliamo di utilizzare i dashboard di S3 Storage Lens anziché utilizzare i parametri di Amazon. CloudWatch Nella dashboard di Storage Lens, puoi visualizzare le seguenti metriche, che monitorano il numero o la dimensione degli oggetti:

- Byte della versione corrente
- Conteggio oggetti versione corrente
- Byte di versione non correnti
- Conteggio di oggetti versione non corrente
- Conteggio oggetti contrassegno di eliminazione
- Byte di archiviazione dei contrassegni di eliminazione
- Byte con caricamento in più parti incompleto
- Conteggio di oggetti con caricamento in più parti incompleto

Il numero di oggetti S3 continua ad aumentare, anche dopo aver impostato le regole del ciclo di vita su un bucket abilitato al controllo delle versioni.

In un bucket abilitato al rilascio di versioni, quando un oggetto è scaduto, l'oggetto non viene eliminato completamente dal bucket. Come versione più recente dell'oggetto viene invece creato un contrassegno di eliminazione. I contrassegni di eliminazione vengono comunque conteggiati come oggetti. Pertanto, se viene creata una regola del ciclo di vita per far scadere solo le versioni correnti, il numero di oggetti nel bucket S3 aumenta anziché diminuire.

Ad esempio, supponiamo che un bucket S3 con il controllo delle versioni abilitato contenga 100 oggetti e che una regola del ciclo di vita sia impostata per far scadere le versioni correnti dell'oggetto dopo 7 giorni. Dopo il settimo giorno, il numero di oggetti aumenta a 200 perché vengono creati 100 contrassegni di eliminazione oltre ai 100 oggetti originali, che ora sono versioni non correnti. Per ulteriori informazioni sulle azioni delle regole di configurazione del ciclo di vita S3 per i bucket con il controllo delle versioni abilitato, consulta Impostazione della configurazione del ciclo di vita in un bucket.

Per rimuovere definitivamente gli oggetti, aggiungi un'ulteriore configurazione del ciclo di vita per eliminare le versioni precedenti degli oggetti, i contrassegni di eliminazione scaduti e i caricamenti incompleti in più parti. Per istruzioni su come creare nuove regole del ciclo di vita, consulta Impostazione della configurazione del ciclo di vita in un bucket.

Note

- Amazon S3 arrotonda la data di transizione o scadenza di un oggetto alla mezzanotte UTC del giorno successivo.

  Quando valuta gli oggetti per le azioni del ciclo di vita, Amazon S3 utilizza l'ora di creazione degli oggetti in UTC. Ad esempio, considera un bucket senza versioni con una regola del ciclo di vita configurata per far scadere gli oggetti dopo un giorno. Supponiamo che un oggetto sia stato creato il 1° gennaio alle 17:05 Pacific Daylight Time (PDT), che corrispondono al 2 gennaio alle 00:05 UTC. L'oggetto diventa vecchio di un giorno alle 00:05 UTC del 3 gennaio, il che lo rende idoneo alla scadenza quando S3 Lifecycle valuta gli oggetti alle 00:00 UTC del 4 gennaio.

  Poiché le azioni del ciclo di vita di Amazon S3 avvengono in modo asincrono, potrebbe verificarsi un certo ritardo tra la data specificata nella regola del ciclo di vita e l'effettiva transizione fisica dell'oggetto. Per ulteriori informazioni, consulta Transizione o ritardo di scadenza.

  Per ulteriori informazioni, consulta Regole del ciclo di vita basate sull'età di un oggetto.
- Per gli oggetti S3 protetti dalla funzionalità Blocco oggetti, le versioni correnti non vengono eliminate definitivamente. Agli oggetti viene invece aggiunto un contrassegno di eliminazione, che li rende non correnti. Le versioni non correnti vengono quindi conservate e non impostate come definitivamente scadute.

Come posso svuotare il mio bucket S3 utilizzando le regole del ciclo di vita?

Le regole del ciclo di vita S3 sono uno strumento efficace per svuotare un bucket S3 contenente milioni di oggetti. Per eliminare un gran numero di oggetti dal tuo bucket S3, assicurati di utilizzare queste due coppie di regole del ciclo di vita:

- Impostazione come scadute delle versioni correnti degli oggetti ed eliminazione definitiva delle versioni precedenti degli oggetti
- Eliminazione dei contrassegni di eliminazione ed eliminazione dei caricamenti in più parti incompleti

Per la procedura di creazione di una regola di configurazione del ciclo di vita, consulta Impostazione della configurazione del ciclo di vita in un bucket.

> **Note**
>
> Per gli oggetti S3 protetti dalla funzionalità Blocco oggetti, le versioni correnti non vengono eliminate definitivamente. Agli oggetti viene invece aggiunto un contrassegno di eliminazione, che li rende non correnti. Le versioni non correnti vengono quindi conservate e non impostate come definitivamente scadute.

La mia fattura Amazon S3 è aumentata dopo la transizione degli oggetti a una classe di archiviazione con costi inferiori.

Esistono diversi motivi per cui la fattura potrebbe aumentare dopo la transizione degli oggetti a una classe di archiviazione con costi inferiori:

- Costo generale S3 Glacier per oggetti di piccole dimensioni

  Per ogni oggetto passato alla classe di archiviazione Recupero flessibile Amazon S3 Glacier o Deep Archive Amazon S3 Glacier, a questo aggiornamento dell'archiviazione è associato un costo generale aggiuntivo di 40 KB. Come parte del costo generale di 40 KB, 8 KB vengono utilizzati per archiviare i metadati e il nome dell'oggetto. Questi 8 KB vengono addebitati in base alle tariffe della classe di archiviazione Amazon S3 Standard. I restanti 32 KB vengono utilizzati per l'indicizzazione e i relativi metadati. Questi 32 KB vengono addebitato in base ai prezzi della classe di archiviazione Recupero flessibile Amazon S3 Glacier o Deep Archive Amazon S3 Glacier.

  Pertanto, se si archiviano molti oggetti di dimensioni più piccole, non è consigliabile utilizzare le transizioni del ciclo di vita. Per ridurre eventuali costi aggiuntivi, valuta la possibilità di aggregare diversi oggetti di piccole dimensioni in un numero più contenuto di oggetti di grandi dimensioni prima di eseguirne l'archiviazione in Amazon S3. Per ulteriori informazioni sulle considerazioni relative ai costi, consulta l'argomento relativo al trasferimento nelle classi di archiviazione Recupero flessibile Amazon S3 Glacier e Deep Archive Amazon S3 Glacier (archiviazione di oggetti).
- Costi minimi di archiviazione

  Alcune classi di archiviazione S3 hanno requisiti minimi di durata dell'archiviazione. Agli oggetti eliminati, sovrascritti o sottoposti a transizione da tali classi prima del raggiungimento della durata minima viene addebitata una tariffa di transizione o eliminazione anticipata proporzionale. Questi requisiti minimi di durata dell'archiviazione sono i seguenti:

- Accesso Infrequente Amazon S3 Standard (AI S3 Standard) e Accesso infrequente a zona unica Amazon S3 (AI a zona unica S3): 30 giorni
- Recupero flessibile Amazon S3 Glacier e Recupero instantaneo Amazon S3 Glacier: 90 giorni
- Deep Archive Amazon S3 Glacier: 180 giorni

Per ulteriori informazioni su questi requisiti, consulta la sezione Vincoli dell'argomento Trasferimento degli oggetti utilizzando il ciclo di vita S3. Per informazioni generali sui prezzi di S3, consulta Prezzi di Amazon S3 e il Calcolatore dei prezzi AWS.

- Costi delle transizioni del ciclo di vita

  Ogni volta che un oggetto viene trasferito a una classe di archiviazione diversa mediante una regola del ciclo di vita, Amazon S3 considera tale transizione come un'unica richiesta di transizione. I costi per queste richieste di transizione si aggiungono ai costi validi per le classi di archiviazione in questione. Se si ha intenzione di trasferire un numero elevato di oggetti, è consigliabile considerare i costi delle transizioni richieste in caso di transizione a una classe inferiore. Per ulteriori informazioni, consulta i Prezzi di Amazon S3.

Ho aggiornato la mia policy di bucket, ma i miei oggetti S3 vengono ancora eliminati a causa delle regole del ciclo di vita scadute.

Le istruzioni `Deny` in una policy di bucket non impediscono la scadenza degli oggetti definiti in una regola del ciclo di vita. Le operazioni del ciclo di vita (come transizioni o scadenze) non utilizzano l'operazione S3 `DeleteObject`. Le operazioni del ciclo di vita di S3 vengono invece eseguite utilizzando endpoint S3 interni. Per ulteriori informazioni, consulta Ciclo di vita e registrazione.

Per evitare che la regola del ciclo di vita esegua operazioni, è necessario modificare, eliminare o disabilitare la regola.

Posso recuperare oggetti S3 scaduti in base alle regole del ciclo di vita di S3?

L'unico modo per recuperare gli oggetti scaduti in base al ciclo di vita S3 è tramite il controllo delle versioni, che deve essere attivo prima che gli oggetti diventino idonei alla scadenza. Non è possibile annullare le operazioni di scadenza eseguite dalle regole del ciclo di vita. Se gli oggetti vengono eliminati definitivamente in base alle regole del ciclo di vita S3 applicate, non sarà possibile recuperarli. Per abilitare il controllo delle versioni su un bucket, consulta the section called “Utilizzo della funzione Controllo delle versioni S3”.

Se al bucket è stato applicato il controllo delle versioni e le versioni non correnti degli oggetti sono ancora intatte, è possibile ripristinare le versioni precedenti degli oggetti scaduti. Per ulteriori informazioni sul comportamento delle operazioni delle regole del ciclo di vita S3 e sugli stati del controllo delle versioni, consulta la tabella Operazioni del ciclo di vita e stato della funzione Controllo delle versioni nel bucket in Elementi per la descrizione delle operazioni nel ciclo di vita.

> **Note**
>
> Se il bucket S3 è protetto da Backup AWS o Replica Amazon S3, è anche possibile utilizzare queste funzionalità per recuperare gli oggetti scaduti.

Risoluzione dei problemi nella replica

Questa sezione riporta i suggerimenti per la risoluzione dei problemi di Replica Amazon S3 e informazioni sugli errori di replica in batch di Amazon S3.

Argomenti

- Suggerimenti per la risoluzione dei problemi di Replica Amazon S3
- Errori di replica in batch

Suggerimenti per la risoluzione dei problemi di Replica Amazon S3

Se le repliche degli oggetti non vengono visualizzate nel bucket di destinazione dopo aver configurato la replica, usa questi suggerimenti per identificare e correggere i problemi.

- La replica della maggior parte degli oggetti viene eseguita entro 15 minuti. Il tempo impiegato da Amazon S3 per replicare un oggetto dipende da diversi fattori, tra cui la combinazione di regione di origine/regione di destinazione e le dimensioni dell'oggetto. La replica di oggetti di grandi dimensioni può richiedere anche diverse ore. Per una migliore visibilità dei tempi di replica, puoi utilizzare la funzionalità di controllo del tempo di replica di S3 (S3 RTC).

  Se l'oggetto replicato è di grandi dimensioni, attendi qualche minuto prima di controllare se è diventato disponibile nella destinazione. È inoltre possibile controllare lo stato della replica dell'oggetto di origine. Se lo stato della replica dell'oggetto è `PENDING`, Amazon S3 non ha completato la replica. Se lo stato della replica dell'oggetto è `FAILED`, controlla la configurazione della replica impostata nel bucket di origine. Inoltre, per ricevere informazioni sugli errori di Amazon

S3 durante la replica, è possibile configurare la funzionalità Notifiche eventi Amazon S3 in modo da ricevere gli eventi di errore di replica. Per ulteriori informazioni, consulta Ricezione di eventi di errore di replica con notifiche di eventi Amazon S3.

- È possibile richiamare l'operazione API `HeadObject` per verificare lo stato della replica un oggetto. L'operazione API `HeadObject` restituisce lo stato della replica `PENDING`, `COMPLETED` o `FAILED` di un oggetto. In risposta a una chiamata API `HeadObject`, lo stato della replica viene restituito nell'elemento `x-amz-replication-status`.

  > **Note**
  >
  > Per eseguire `HeadObject`, è necessario disporre dell'accesso in lettura all'oggetto che si sta richiedendo. Una richiesta `HEAD` ha le stesse opzioni di una richiesta `GET`, senza eseguire alcuna operazione `GET`. Ad esempio, per eseguire una richiesta `HeadObject` utilizzando la AWS Command Line Interface (AWS CLI), puoi eseguire il seguente comando. Sostituire *`user input placeholders`* con le proprie informazioni.
  >
  > ```
  > aws s3api head-object --bucket my-bucket --key index.html
  > ```

- Dopo la restituzione da parte di `HeadObject` degli oggetti con uno stato della replica `FAILED`, puoi utilizzare Replica Amazon S3 per eseguire la replica degli oggetti con replica non riuscita. In alternativa, puoi caricare nuovamente gli oggetti con replica non riuscita nel bucket di origine, che avvierà la replica dei nuovi oggetti.
- Nella configurazione di replica nel bucket di origine verifica quanto segue:
  - La correttezza dell'Amazon Resource Name (ARN) relativo al bucket di destinazione.
  - La correttezza del prefisso del nome della chiave. Ad esempio, se si imposta la configurazione per replicare gli oggetti con il prefisso `Tax`, solo gli oggetti con i nomi della chiave quali `Tax/document1` o `Tax/document2` vengono replicati. Un oggetto con il nome della chiave `document3` non sia replicato.
  - Lo stato della regola di replica è `Enabled`.
- Verifica che il controllo delle versioni non sia stato sospeso per i bucket inclusi nella configurazione della replica. Sia per il bucket di origine che per quello di destinazione deve essere abilitata la funzione Controllo delle versioni.
- Se una regola di replica è impostata su Assegna la proprietà degli oggetti al proprietario del bucket di destinazione, il ruolo AWS Identity and Access Management (IAM) utilizzato per la replica deve disporre dell'autorizzazione `s3:ObjectOwnerOverrideToBucketOwner`. Questa autorizzazione

viene concessa sulla risorsa (in questo caso, il bucket di destinazione). Ad esempio, la seguente istruzione `Resource` mostra come concedere questa autorizzazione al bucket di destinazione:

```
{
  "Effect":"Allow",
  "Action":[
    "s3:ObjectOwnerOverrideToBucketOwner"
  ],
  "Resource":"arn:aws:s3:::DestinationBucket/*"
}
```

- Se il bucket di destinazione è di proprietà di un altro account, il proprietario del bucket di origine deve concedere l'autorizzazione `s3:ObjectOwnerOverrideToBucketOwner` al proprietario del bucket di origine. Per utilizzare la seguente policy di esempio, sostituisci *`user input placeholders`* con le tue informazioni.

```
{
  "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**
>
> Se le impostazioni dell'opzione Proprietà dell’oggetto del bucket di destinazione includono Bucket owner enforced, non è necessario aggiornare l'impostazione su Assegna la proprietà degli oggetti al proprietario del bucket di destinazione nella regola di replica. La

modifica della proprietà dell'oggetto avverrà per impostazione predefinita. Per ulteriori informazioni sulla modifica della proprietà della replica, consulta Modifica del proprietario della replica.

- Se stai impostando la configurazione di replica in uno scenario tra più account, in cui i bucket di origine e di destinazione sono di proprietà di diversiAccount AWS, i bucket di destinazione non possono essere configurati come bucket Requester Pays. Per ulteriori informazioni, consulta Utilizzo dei bucket con pagamento a carico del richiedente per utilizzo e trasferimenti di storage.
- Se gli oggetti di origine gente di un bucket sono crittografati con una chiave AWS Key Management Service (AWS KMS), la regola di replica deve essere configurata per includere oggetti con crittografia AWS KMS. Assicurati di selezionare Replica oggetti crittografati con AWS KMS nelle impostazioni dell'opzione Crittografia nella console Amazon S3. Seleziona quindi una chiave AWS KMS per crittografare gli oggetti di destinazione.

> **Note**
>
> Se il bucket di destinazione si trova in un account diverso, specifica una chiave gestita dal cliente AWS KMS di proprietà dell'account di destinazione. Non utilizzare la chiave predefinita gestita da Amazon S3 (`aws/s3`). L'utilizzo della chiave predefinita crittografa gli oggetti con la chiave gestita da Amazon S3 di proprietà dell'account di origine, impedendo che l'oggetto venga condiviso con un altro account. Di conseguenza, l'account di destinazione non sarà in grado di accedere agli oggetti nel bucket di destinazione.

Per utilizzare una chiave AWS KMS appartenente all'account di destinazione per crittografare gli oggetti di destinazione, l'account di destinazione deve concedere le autorizzazioni `kms:GenerateDataKey` e `kms:Encrypt` al ruolo di replica nella policy della chiave KMS. Per utilizzare la seguente istruzione di esempio nella policy della chiave KMS, sostituisci *`user input placeholders`* con le tue informazioni:

```
{
    "Sid": "AllowS3ReplicationSourceRoleToUseTheKey",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::123456789101:role/s3-replication-role"
    },
    "Action": ["kms:GenerateDataKey", "kms:Encrypt"],
    "Resource": "*"
```

```
}
```

Se usi un asterisco (*) per l'istruzione `Resource` nella policy della chiave AWS KMS, la policy concede l'autorizzazione all'uso della chiave KMS solo per il ruolo di replica. La policy non consente al ruolo di replica di aumentare il livello delle proprie autorizzazioni.

Per impostazione predefinita, la policy della chiave KMS concede all'utente root le autorizzazioni complete per la chiave. Queste autorizzazioni possono essere delegate ad altri utenti nello stesso account. A meno che non siano presenti istruzioni `Deny` nella policy della chiave KMS di origine, è sufficiente utilizzare una policy IAM per concedere le autorizzazioni del ruolo di replica alla chiave KMS di origine.

> **Note**
>
> Le policy della chiave KMS che limitano l'accesso a intervalli CIDR, endpoint VPC o punti di accesso S3 specifici possono causare la mancata riuscita della replica.

Se le chiavi KMS di origine o destinazione concedono autorizzazioni in base al contesto di crittografia, verifica che le chiavi dei bucket Amazon S3 siano attivate per i bucket. Se le chiavi dei bucket Amazon S3 sono attive per i bucket, il contesto di crittografia deve essere la risorsa a livello di bucket, come segue:

```
"kms:EncryptionContext:arn:aws:arn": [
     "arn:aws:s3:::SOURCE_BUCKET_NAME"
     ]
"kms:EncryptionContext:arn:aws:arn": [
     "arn:aws:s3:::DESTINATION_BUCKET_NAME"
     ]
```

Oltre alle autorizzazioni concesse dalla policy della chiave KMS, l'account di origine deve aggiungere le seguenti autorizzazioni minime alla policy IAM del ruolo di replica:

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
```

```
    "Resource": [
        "SourceKmsKeyArn"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "kms:GenerateDataKey",
        "kms:Encrypt"
    ],
    "Resource": [
        "DestinationKmsKeyArn"
    ]
}
```

Per ulteriori informazioni su come eseguire la replica di oggetti crittografati con AWS KMS, consulta Replica di oggetti crittografati.

- Se il bucket di destinazione è di proprietà di un altro Account AWS, verifica che il proprietario di tale bucket disponga di una policy del bucket che consenta al proprietario del bucket di origine di replicare gli oggetti. Per vedere un esempio, consulta Configurazione della replica quando i bucket di origine e di destinazione sono di proprietà di account diversi.
- Se i tuoi oggetti non si replicano anche dopo aver convalidato le autorizzazioni, verifica la presenza di eventuali istruzioni `Deny` esplicite nelle seguenti posizioni:
  - Le istruzioni `Deny` nelle policy di bucket di origine o di destinazione. La replica non riesce se la policy del bucket nega l'accesso al ruolo di replica per una delle seguenti operazioni:

    Bucket di origine:

```
        "s3:GetReplicationConfiguration",
        "s3:ListBucket",
        "s3:GetObjectVersionForReplication",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
```

    Bucket di destinazione:

```
        "s3:ReplicateObject",
```

```
            "s3:ReplicateDelete",
            "s3:ReplicateTags"
```

  - Le istruzioni `Deny` o i limiti delle autorizzazione associati al ruolo IAM possono causare la mancata esecuzione della replica.
  - Le istruzioni `Deny` contenute nelle policy di controllo del servizio AWS Organizations associate agli account di origine o di destinazione possono causare la mancata esecuzione della replica.
- Se la replica di un oggetto non è presente nel bucket di destinazione, il problema a livello di replica potrebbe essere dovuto alle cause seguenti:
  - Amazon S3 non replica un oggetto in un bucket di origine che è una replica creata da un'altra configurazione di replica. Se, ad esempio, imposti una configurazione di replica dal bucket A al bucket B al bucket C, Amazon S3 non replica le repliche degli oggetti del bucket B nel bucket C.
  - Il proprietario di un bucket di origine può concedere ad altri Account AWS l'autorizzazione necessaria per caricare gli oggetti. Per impostazione predefinita, il proprietario del bucket di origine non dispone di autorizzazioni per gli oggetti creati da altri account. La configurazione di replica esegue la replica solo degli oggetti per i quali il proprietario del bucket di origine dispone delle autorizzazioni di accesso. Il proprietario del bucket di origine può concedere ad altri Account AWS le autorizzazioni necessarie per creare oggetti in modo condizionale, richiedendo autorizzazioni di accesso esplicite su quegli oggetti. Per un esempio di policy, consulta Concedere autorizzazioni multi-account per il caricamento di oggetti a garanzia del controllo completo da parte del proprietario del bucket.
- Supponiamo di aggiungere nella configurazione della replica una regola per replicare un sottoinsieme di oggetti con un tag specifico. In questo caso, è necessario assegnare il valore e la chiave del tag specifici al momento della creazione dell'oggetto per permettere ad Amazon S3 di replicare l'oggetto. Se prima crei un oggetto e quindi aggiungi il tag all'oggetto esistente, Amazon S3 non replica l'oggetto.
- Usa la funzionalità Notifiche eventi Amazon S3 per inviare un avviso nei casi in cui gli oggetti non vengono replicati nella Regione AWS di destinazione. Le notifiche eventi di Amazon S3 sono disponibili tramite Amazon Simple Queue Service (Amazon SQS), Servizio di notifica semplice Amazon (Amazon SNS) o AWS Lambda. Per ulteriori informazioni, consulta Ricezione di eventi di errore di replica con notifiche di eventi Amazon S3.

  Puoi anche visualizzare i motivi degli errori di replica usando la funzionalità Notifiche eventi Amazon S3. Per esaminare l'elenco dei motivi degli errori, consulta Motivi degli errori di replica Amazon S3.

Errori di replica in batch

Per risolvere i problemi relativi agli oggetti che non vengono replicati nel bucket di destinazione, controlla i diversi tipi di autorizzazioni per il bucket, il ruolo di replica e il ruolo IAM utilizzati per creare il processo di replica in batch. Inoltre, assicurati di controllare le impostazioni di accesso pubblico e le impostazioni della proprietà del bucket.

Durante l'utilizzo della replica in batch, è possibile che si verifichi uno dei seguenti errori:

- Lo stato dell'operazione in batch non è riuscito e il motivo è: non è stato possibile scrivere il report del processo nel bucket dei report.

  Questo errore si verifica se il ruolo IAM utilizzato per il processo Operazioni in batch non è in grado di inserire il report di completamento nella posizione specificata al momento della creazione del processo. Per risolvere questo problema, verifica che il ruolo IAM disponga delle autorizzazioni `PutObject` per il bucket in cui desideri salvare il report di completamento delle operazioni in batch. Si tratta di una best practice per inviare il report a un bucket diverso da quello di origine.
- L'operazione in batch è stata completata con errori e il totale degli errori non è 0.

  Questo errore si verifica in presenza di problemi relativi ad autorizzazioni oggetti insufficienti per il processo di replica in batch in esecuzione. Se utilizzi una regola di replica per il processo di replica in batch, assicurati che il ruolo IAM utilizzato per la replica disponga delle autorizzazioni appropriate per accedere agli oggetti dal bucket di origine o di destinazione. Puoi anche controllare il rapporto sul completamento della replica in batch per esaminare il motivo specifico dell'errore di Replica Amazon S3.
- Il processo batch è stato eseguito correttamente ma il numero di oggetti previsti nel bucket di destinazione non è lo stesso.

  Questo errore si verifica quando c'è una mancata corrispondenza tra gli oggetti elencati nel manifesto fornito nel processo di replica in batch e i filtri selezionati al momento della creazione del processo. È possibile che questo messaggio venga visualizzato anche quando gli oggetti nel bucket di origine non corrispondono a nessuna regola di replica e non sono inclusi nel manifesto generato.

Risoluzione dei problemi di registrazione degli accessi al server

Gli argomenti seguenti possono essere utili per risolvere i problemi che possono verificarsi durante la configurazione della registrazione con Amazon S3.

Argomenti

Messaggi di errore comuni durante la configurazione della registrazione

I seguenti messaggi di errore comuni possono essere visualizzati quando si abilita la registrazione tramite la AWS Command Line Interface (AWS CLI) e gli SDK AWS:

Errore: Registrazione multi-posizione S3 non consentita

Se il bucket di destinazione (noto anche come bucket target) si trova in una regione diversa da quella del bucket di origine, si verifica l'errore Registrazione multi-posizione S3 non consentita. Per risolvere questo errore, verifica che il bucket di destinazione configurato per ricevere i log degli accessi si trovi nella stessa Regione AWS e nello stesso Account AWS del bucket di origine.

Errore: Il proprietario del bucket da registrare e quello del bucket di destinazione devono coincidere

Quando si abilita la registrazione degli accessi al server, questo errore si verifica se il bucket di destinazione specificato appartiene a un account diverso. Per risolvere questo problema, verifica che il bucket di destinazione si trovi nello stesso Account AWS del bucket di origine.

> **Note**
>
> Ti consigliamo di scegliere un bucket di destinazione diverso dal bucket di origine. Quando il bucket di origine e il bucket di destinazione coincidono, vengono creati log aggiuntivi per i log scritti nel bucket. Questa situazione può causare l'aumento delle spese di archiviazione. La registrazione di questi log aggiuntivi può rendere difficile la ricerca di log specifici. Tuttavia, per una gestione più semplice dei log, si consiglia di salvare i log degli accessi in un bucket diverso. Per ulteriori informazioni, consulta the section called “Come si abilita il recapito dei log?”.

Errore: Il bucket di destinazione per la registrazione non esiste

Il bucket di destinazione deve esistere prima di impostare la configurazione. Questo errore indica che il bucket di destinazione non esiste o non può essere trovato. Verifica che il nome del bucket sia stato digitato correttamente, quindi riprova.

Errore: Concessioni di destinazione non consentite per i bucket con l'opzione Bucket owner enforced abilitata

Questo errore indica che il bucket di destinazione utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto S3. L'impostazione Proprietario del bucket applicato non supporta le concessioni di destinazione (target). Per ulteriori informazioni, consulta Autorizzazioni per la distribuzione dei registri.

Risoluzione dei problemi di consegna

Per evitare problemi di registrazione degli accessi al server, assicurati di seguire queste best practice:

- Il gruppo di distribuzione dei log S3 dispone dell'accesso in scrittura al bucket di destinazione: il gruppo di distribuzione dei log S3 fornisce i log degli accessi al server al bucket di destinazione. Puoi usare una policy del bucket o una lista di controllo degli accessi (ACL) di un bucket per concedere l'accesso in scrittura al bucket di destinazione. Invece di una lista di controllo degli accessi (ACL) consigliamo di utilizzare una policy di bucket. Per ulteriori informazioni su come concedere l'accesso in scrittura al bucket di destinazione, consulta Autorizzazioni per la distribuzione dei registri.

  > **Note**
  >
  > Se il bucket di destinazione utilizza l'impostazione Proprietario del bucket applicato per Proprietà dell'oggetto, tieni presente quanto segue:
  >
  > - Le ACL sono disabilitate e non hanno più alcuna ripercussione sulle autorizzazioni. Ciò significa che non è possibile aggiornare l'ACL del bucket per concedere l'accesso al gruppo di distribuzione dei log S3. Invece, è necessario aggiornare la policy del bucket per il bucket di destinazione per concedere l'accesso al principale del servizio di registrazione.
  > - Non puoi includere concessioni di destinazione nella configurazione `PutBucketLogging`.

- La policy del bucket per il bucket di destinazione consente l'accesso ai log: controlla la policy del bucket del bucket di destinazione. Nella policy di bucket cerca tutte le istruzioni contenenti `"Effect": "Deny"`. Verifica quindi che l'istruzione `Deny` non impedisca la scrittura dei log di accesso nel bucket.
- S3 Object Lock non è abilitato sul bucket di destinazione: controlla se per il bucket di destinazione è stata abilitata l'opzione Object Lock. L'opzione Blocco oggetti blocca la consegna del log degli

accessi al server. È necessario scegliere un bucket di destinazione in cui non sia abilitata l'opzione Object Lock.

- Le chiavi gestite da Amazon S3 (SSE-S3) vengono selezionate se la crittografia predefinita è abilitata sul bucket di destinazione: puoi usare la crittografia bucket predefinita sul bucket di destinazione solo se utilizzi la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3). La crittografia lato server predefinita con chiavi AWS Key Management Service (AWS KMS) (SSE-KMS) non è supportata per i bucket di destinazione con la registrazione degli accessi al server. Per ulteriori informazioni su come abilitare la crittografia predefinita, consulta Configurazione della crittografia predefinita.
- Nel bucket di destinazione l'opzione Pagamento a carico del richiedente non è abilitata: non è supportato l'uso di un bucket con pagamento a carico del richiedente come bucket di destinazione per la registrazione degli accessi al server. Per consentire la consegna dei log di accesso ai server, disabilita l'opzione Pagamento a carico del richiedente nel bucket di destinazione.
- Controlla la policy di controllo del servizio AWS Organizations: durante l'utilizzo di AWS Organizations, controlla le policy di controllo del servizio per assicurarti che l'accesso ad Amazon S3 sia consentito. Le policy di controllo del servizio specificano le autorizzazioni massime per gli account interessati. Nella policy di controllo del servizio cerca tutte le istruzioni contenenti `"Effect": "Deny"` e verifica che le istruzioni `Deny` non impediscano la scrittura dei log degli accessi nel bucket. Per ulteriori informazioni, consulta Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations.
- Attendi qualche minuto affinché le modifiche recenti alla configurazione dei log siano effettive: l'abilitazione della registrazione di accesso ai server per la prima volta o la modifica del bucket di destinazione per i log richiede tempo per essere pienamente effettiva. Potrebbe essere necessaria più di un'ora prima che tutte le richieste vengano registrate e consegnate correttamente.

  Per verificare eventuali errori di consegna dei log, abilita le metriche delle richieste in Amazon. CloudWatch Se i log non vengono consegnati entro poche ore, cerca la metrica `4xxErrors`, che può indicare gli errori di consegna dei log. Per ulteriori informazioni sull'abilitazione delle metriche delle richieste, consulta the section called “Creazione di una configurazione di parametri per tutti gli oggetti”.

Risoluzione dei problemi relativi al controllo delle versioni

I seguenti argomenti sono utili per risolvere alcuni problemi comuni relativi alla funzione Controllo delle versioni di Amazon S3.

Argomenti

Desidero recuperare oggetti che sono stati eliminati per errore in un bucket in cui la funzione Controllo delle versioni è abilitata

In generale, quando le versioni degli oggetti vengono eliminate dai bucket S3, Amazon S3 non può più recuperarle. Tuttavia, se hai abilitato la funzione S3 Controllo delle versioni sul tuo bucket S3, una richiesta `DELETE` che non specifica un ID di versione non può eliminare definitivamente un oggetto. Viene invece aggiunto un contrassegno di eliminazione come segnaposto. Questo contrassegno di eliminazione diventa la versione corrente dell'oggetto.

Per verificare se gli oggetti eliminati sono rimossi definitivamente o temporaneamente (sostituiti da un contrassegno di eliminazione), procedi come segue:

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nel pannello di navigazione a sinistra, scegli Buckets (Bucket).
3. Nell'elenco Buckets (Bucket) scegliere il nome del bucket contenente l'oggetto.
4. Nell'elenco Oggetti, attiva il controllo Mostra versioni a destra della barra di ricerca, quindi cerca l'oggetto eliminato nella barra di ricerca. Questo controllo è disponibile solo se la funzione Controllo delle versioni è stata precedentemente abilitata nel bucket.

   Puoi anche utilizzare S3 Inventory per cercare gli oggetti eliminati.
5. Se non riesci a trovare l'oggetto dopo aver attivato il controllo Mostra versioni o dopo avere creato un report di inventario non riesci a trovare il contrassegno di eliminazione dell'oggetto, significa che l'eliminazione è permanente e l'oggetto non può più essere recuperato.

Puoi anche verificare lo stato di un oggetto eliminato utilizzando l'operazione `HeadObject` API di AWS Command Line Interface (AWS CLI). A tale scopo, utilizza il comando `head-object` e sostituisci *`user input placeholders`* con le tue informazioni.

```
aws s3api head-object --bucket example-s3-bucket --key index.html
```

Se esegui il comando `head-object` su un oggetto con il controllo delle versioni abilitato la cui versione corrente è un contrassegno di cancellazione, riceverai un errore 404 Non trovato. Per esempio:

Si è verificato un errore (404) durante la chiamata dell' HeadObject operazione: Not Found

Se esegui il comando `head-object` su un oggetto con il controllo delle versioni abilitato e fornisci l'ID versione dell'oggetto, Amazon S3 recupera i metadati dell'oggetto, a conferma che l'oggetto esiste ancora e che non è stato eliminato definitivamente.

```
aws s3api head-object --bucket example-s3-bucket --key index.html --
version-id versionID
```

```
{
"AcceptRanges": "bytes",
"ContentType": "text/html",
"LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
"ContentLength": 77,
"VersionId": "Zg5HyL7m.eZU9iM7AVlJkrqAiE.0UG4q",
"ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
"Metadata": {}
}
```

Se l'oggetto viene trovato e la relativa versione più recente è un contrassegno di eliminazione, significa che la versione precedente dell'oggetto esiste ancora. Poiché il contrassegno di eliminazione corrisponde alla versione corrente dell'oggetto, è possibile recuperare l'oggetto eliminando il contrassegno di eliminazione.

Dopo aver rimosso definitivamente il contrassegno di eliminazione, la seconda versione più recente dell'oggetto diventa la versione corrente, rendendolo nuovamente disponibile. Per una rappresentazione visiva di come gli oggetti vengono recuperati, consulta Rimozione dei contrassegni di eliminazione.

Per rimuovere una versione specifica di un oggetto, devi essere il proprietario del bucket. Per eliminare definitivamente un contrassegno di eliminazione occorre includere il suo ID versione nella richiesta `DeleteObject`. Per eliminare il contrassegno di eliminazione, usa il seguente comando e sostituisci *`user input placeholders`* con le tue informazioni:

```
aws s3api delete-object --bucket example-s3-bucket --key index.html --
version-id versionID
```

Per ulteriori informazioni sul comando `delete-object`, consulta delete-object in the Guida di riferimento ai comandi della AWS CLI . Per informazioni sull'eliminazione definitiva dei contrassegni di eliminazione, consulta Gestione dei contrassegni di eliminazione.

Voglio eliminare definitivamente gli oggetti con il controllo delle versioni abilitato

In un bucket con la funzione Controllo delle versioni abilitata, una richiesta `DELETE` senza un ID versione non può eliminare un oggetto definitivamente. Questo tipo di richiesta inserisce invece un contrassegno di eliminazione.

Per eliminare oggetti con la funzione Controllo delle versioni abilitata in modo permanente, puoi scegliere tra i seguenti metodi:

- Crea una regola del ciclo di vita S3 per eliminare definitivamente le versioni non correnti. Per eliminare in modo definitivo le versioni non correnti, in Elimina in modo definitivo le versioni non correnti degli oggetti, specifica il numero di giorni nel campo Numero di giorni dopo il quale gli oggetti diventano non correnti. Facoltativamente puoi specificare il numero di versioni più recenti da mantenere immettendo un valore nel campo Number of newer versions to retain (Numero di versioni più recenti da mantenere). Per ulteriori informazioni sulla creazione di questa regola, consulta l'argomento relativo all'impostazione di una configurazione del ciclo di vita S3.
- Elimina una versione specificata includendo l'ID versione nella richiesta `DELETE`. Per ulteriori informazioni, consulta l'argomento relativo alla procedura di eliminazione definitiva degli oggetti con controllo delle versioni abilitata.
- Crea una regola del ciclo di vita per far scadere le versioni correnti. Per definire la scadenza delle versioni correnti degli oggetti, seleziona Scadenza versioni correnti degli oggetti e aggiungi un numero in Giorni dopo la creazione degli oggetti. Per ulteriori informazioni sulla creazione di questa regola del ciclo di vita, consulta l'argomento relativo all'impostazione di una configurazione del ciclo di vita S3.
- Per eliminare definitivamente tutti gli oggetti con il controllo delle versioni abilitato e i relativi contrassegni di eliminazione, crea due regole del ciclo di vita: una per far scadere le versioni correnti ed eliminare definitivamente le versioni non correnti degli oggetti e l'altra per eliminare i contrassegni di eliminazione degli oggetti scaduti.

In un bucket con il controllo delle versioni abilitato, una richiesta DELETE che non specifica un ID versione può rimuovere solo gli oggetti con un ID versione NULL. Se l'oggetto è stato caricato quando il controllo delle versioni era abilitato, una richiesta DELETE che non specifica un ID versione crea un contrassegno di eliminazione per tale oggetto.

> **Note**
>
> Per i bucket con la funzione S3 Blocco oggetti abilitata, una richiesta oggetto DELETE con un ID versione dell'oggetto protetto causa un errore 403 Accesso negato. Una richiesta oggetto DELETE senza un ID versione aggiunge un contrassegno di eliminazione come versione più recente dell'oggetto con una risposta 200 OK. Gli oggetti protetti dalla funzionalità Blocco oggetti non possono essere eliminati definitivamente finché i relativi periodi di conservazione e blocchi a fini legali non vengono rimossi. Per ulteriori informazioni, consulta the section called “Come funziona il blocco oggetti S3”.

Sto riscontrando un peggioramento delle prestazioni dopo aver abilitato il controllo delle versioni del bucket

Il peggioramento delle prestazioni può verificarsi nei bucket con il controllo delle versioni abilitato se sono presenti troppi contrassegni di eliminazione o oggetti con versioni e se non vengono seguite le best practice.

Numero eccessivo di contrassegni di eliminazione

Dopo aver abilitato la funzione Controllo delle versioni in un bucket, una richiesta DELETE senza un ID versione effettuata su un oggetto crea un contrassegno di eliminazione con un ID versione univoco. Le configurazioni del ciclo di vita con una regola Scadenza versioni correnti degli oggetti aggiungono un contrassegno di eliminazione con un ID versione univoco a ogni oggetto. Un numero eccessivo di contrassegni di eliminazione può ridurre le prestazioni nel bucket.

Se la funzione Controllo delle versioni viene disabilitata in un bucket, Amazon S3 contrassegna l'ID versione come NULL per i nuovi oggetti creati. Un'operazione di scadenza in un bucket con la funzione Controllo delle versioni disabilitata fa sì che Amazon S3 crei un contrassegno di eliminazione il cui ID versione è NULL. In un bucket con la funzione Controllo delle versioni disabilitata, viene creato un contrassegno di eliminazione NULL per ogni richiesta di eliminazione. Questi contrassegni di eliminazione NULL sono anche chiamati contrassegni di eliminazione degli oggetti scaduti quando tutte le versioni degli oggetti vengono eliminate e rimane solo un unico

contrassegno di eliminazione. In presenza di un numero eccessivo di contrassegni di eliminazione NULL, si verifica un peggioramento delle prestazioni nel bucket.

Numero eccessivo di oggetti con il controllo delle versioni abilitato

Se un bucket con il controllo delle versioni abilitato contiene oggetti con milioni di versioni, può verificarsi un incremento del numero di errori di tipo 503 Servizio non disponibile. Se rilevi un aumento significativo del numero di risposte HTTP 503 Servizio non disponibile ricevute per le richieste oggetto PUT o DELETE in un bucket con il controllo delle versioni abilitato, è possibile che esistano uno o più oggetti nel bucket per i quali sono presenti milioni di versioni. In presenza di oggetti con milioni di versioni, Amazon S3 limita automaticamente le richieste a livello di bucket. La limitazione delle richieste consente di proteggere il bucket dal traffico generato da un numero eccessivo di richieste, che potrebbe potenzialmente impedire la gestione di altre richieste eseguite nello stesso bucket.

Per determinare quali oggetti sono dotati di milioni di versioni, utilizza S3 Inventory. S3 Inventory genera un report che fornisce un elenco di file flat degli oggetti in un bucket. Per ulteriori informazioni, consulta Amazon S3 Inventory.

Per verificare se nel bucket è presente un numero elevato di oggetti con il controllo delle versioni abilitato, utilizza le metriche di S3 Storage Lens per visualizzare i valori pertinenti nei campi Conteggio oggetti versione corrente, Conteggio di oggetti versione non corrente e Conteggio oggetti contrassegno di eliminazione. Per ulteriori informazioni sulle metriche di Storage Lens, consulta Glossario dei parametri di Amazon S3 Storage Lens.

Il team Amazon S3 consiglia ai clienti di analizzare le applicazioni che sovrascrivono ripetutamente lo stesso oggetto, creando potenzialmente milioni di versioni per tale oggetto, al fine di determinare se il funzionamento dell'applicazione corrisponde a quello previsto. Ad esempio, un'applicazione che sovrascrive lo stesso oggetto ogni minuto per una settimana può creare oltre diecimila versioni. Si consiglia di archiviare meno di centomila versioni per ciascun oggetto. Se hai un caso d'uso che richiede milioni di versioni per uno o più oggetti, contatta il AWS Support team per ricevere assistenza nella determinazione di una soluzione migliore.

Best practice

Per evitare problemi di deterioramento delle prestazioni correlati al controllo delle versioni, consigliamo di adottare le seguenti best practice:

- Abilita una regola del ciclo di vita per far scadere le versioni precedenti degli oggetti. Ad esempio, è possibile creare una regola del ciclo di vita per far scadere le versioni non correnti dopo 30

giorni dal passaggio dell'oggetto allo stato non corrente. Puoi anche conservare più versioni non correnti se non desideri eliminarle tutte. Per ulteriori informazioni, consulta l'argomento relativo all'impostazione di una configurazione del ciclo di vita S3.

- Abilita una regola del ciclo di vita per eliminare i contrassegni di eliminazione degli oggetti scaduti a cui non sono associati oggetti dati nel bucket. Per ulteriori informazioni, consulta l'argomento relativo alla rimozione dei contrassegni di eliminazione degli oggetti scaduti.

Per ulteriori best practice per l'ottimizzazione delle prestazioni di Amazon S3, consulta Modelli di progettazione delle best practice.

Ottenere gli ID delle richieste Amazon S3 per AWS Support

Ogni volta che contatti AWS Support perché hai riscontrato errori o comportamenti imprevisti in Amazon S3, devi fornire gli ID di richiesta associati all'azione non riuscita. AWS Support utilizza questi ID di richiesta per risolvere i problemi che stai riscontrando.

Gli ID richiesta vengono forniti in coppia, vengono restituiti in ogni risposta elaborata da Amazon S3 (anche quelle vere) ed è possibile accedervi tramite i log dettagliati. Esistono diversi metodi comuni per ottenere gli ID delle richieste, inclusi i log di accesso di S3 e AWS CloudTrail gli eventi o gli eventi relativi ai dati.

Dopo aver recuperato questi log, copia e conserva questi due valori, perché ti serviranno quando entri in contatto. AWS Support Per informazioni su come contattare AWS Support, consulta Contatti AWS o la AWS Support documentazione.

Utilizzo di HTTP per recuperare gli ID richiesta

È possibile recuperare gli ID richiesta, `x-amz-request-id` e `x-amz-id-2` registrando le parti di una richiesta HTTP prima che raggiunga l'applicazione di destinazione. Esistono diversi strumenti di terze parti che è possibile utilizzare per recuperare i log verbose per le richieste HTTP. Scegli lo strumento che preferisci usare ed eseguilo per rimanere in ascolto sulla porta che gestisce il traffico di Amazon S3 mentre invii un'altra richiesta HTTP Amazon S3.

Per le richieste HTTP, la coppia di ID richiesta sarà simile agli esempi seguenti:

```
x-amz-request-id: 79104EXAMPLEB723
x-amz-id-2: IOWQ4fDEXAMPLEQM+ey7N9WgVhSnQ6JEXAMPLEZb7hSQDASK+Jd1vEXAMPLEa3Km
```

> **Note**
>
> Le richieste HTTPS sono crittografate e nascoste nella maggior parte delle acquisizioni dei pacchetti.

Utilizzo di un browser Web per recuperare gli ID richiesta

La maggiore dei browser Web include strumenti per sviluppatori che puoi usare per visualizzare le intestazioni delle richieste.

Per le richieste basate sul Web che restituiscono un errore, la coppia di ID richiesta sarà simile agli esempi seguenti.

```
<Error><Code>AccessDenied</Code><Message>Access Denied</Message>
<RequestId>79104EXAMPLEB723</RequestId><HostId>IOWQ4fDEXAMPLEQM
+ey7N9WgVhSnQ6JEXAMPLEZb7hSQDASK+Jd1vEXAMPLEa3Km</HostId></Error>
```

Per ottenere la coppia di ID richiesta proveniente da richieste completate, utilizza gli strumenti per sviluppatori del browser per esaminare le intestazioni delle risposte HTTP. Per informazioni sugli strumenti per gli sviluppatori di browser specifici, consulta Risoluzione dei problemi di Amazon S3 - Recupero degli ID richiesta S3 in AWS re:Post.

Utilizzo degli AWS SDK per ottenere gli ID delle richieste

Le sezioni riportate di seguito includono informazioni per la configurazione dei log utilizzando un SDK AWS . Sebbene sia possibile abilitare la registrazione dettagliata in ogni richiesta e risposta, si sconsiglia di abilitarla nei sistemi di produzione in quanto le richieste e le risposte di grandi dimensioni possono causare notevoli rallentamenti in un'applicazione.

Per le richieste AWS SDK, la coppia di ID di richiesta sarà simile agli esempi seguenti.

```
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
```

Utilizzo dell'SDK for Go per ottenere gli ID delle richieste

Puoi configurare la registrazione utilizzando SDK for Go. Per ulteriori informazioni, consulta i metadati di Response nella Guida per sviluppatori SDK for Go V2.

Utilizzo di SDK per PHP per recuperare gli ID richiesta

È possibile configurare la registrazione utilizzando PHP. Per ulteriori informazioni, consulta Come è possibile controllare i dati che vengono trasmessi in rete? nella Guida per gli sviluppatori di AWS SDK for PHP .

Utilizzo di SDK per Java per recuperare gli ID richiesta

È possibile abilitare la registrazione per richieste o risposte specifiche per acquisire e restituire solo le intestazioni rilevanti. A tale scopo, importare la classe `com.amazonaws.services.s3.S3ResponseMetadata`. È quindi possibile archiviare la richiesta in una variabile prima di eseguire la richiesta effettiva. Per recuperare la richiesta o la risposta registrata, chiama `getCachedResponseMetadata(AmazonWebServiceRequest request).getRequestID()`.

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());
```

In alternativa, è possibile utilizzare la registrazione verbose di ogni richiesta e risposta Java. Per ulteriori informazioni, consulta l'argomento relativo alla registrazione dettagliata in rete nella Guida per gli sviluppatori di AWS SDK for Java .

Utilizzo di per AWS SDK for .NET ottenere gli ID delle richieste

È possibile configurare la registrazione con AWS SDK for .NET utilizzando lo strumento di `System.Diagnostics` registrazione integrato. Per ulteriori informazioni, consulta il post del blog per sviluppatori Logging with the AWS SDK for AWS .NET.

> **Note**
>
> Per default, il log restituito contiene solo informazioni sugli errori. Per recuperare gli ID richiesta, è necessario aggiungere `AWSLogMetrics` (e facoltativamente `AWSResponseLogging`) al file di configurazione.

Utilizzo di SDK per Python (Boto3) per recuperare gli ID richiesta

Con AWS SDK for Python (Boto3), puoi registrare risposte specifiche. È possibile utilizzare questa funzione per acquisire solo le intestazioni pertinenti. Il codice seguente mostra come registrare parti della risposta a un file:

```
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'])
```

È inoltre possibile rilevare le eccezioni e registrare le informazioni pertinenti quando viene sollevata un'eccezione. Per ulteriori informazioni, consulta l'argomento relativo alla ricerca di informazioni utili nelle risposte degli errori nella pagina Riferimento API per l'SDK AWS per Python (Boto).

Inoltre, è possibile configurare Boto3 per l'output dei log di debug dettagliati utilizzando il seguente codice:

```
import boto3
boto3.set_stream_logger('', logging.DEBUG)
```

Per ulteriori informazioni, consulta set_stream_logger nella pagina Riferimento API per l'SDK AWS per Python (Boto).

Utilizzo di SDK per Ruby per recuperare gli ID richiesta

È possibile recuperare gli ID richiesta utilizzando SDK per Ruby versione 1, 2 o 3.

- Mediante l'uso di SDK for Ruby - Versione 1: è possibile abilitare la registrazione in rete HTTP a livello globale utilizzando la seguente riga di codice.

```
s3 = AWS::S3.new(:logger => Logger.new($stdout), :http_wire_trace => true)
```

- Mediante l'uso di SDK for Ruby - Versione 2 o Versione 3: è possibile abilitare la registrazione in rete HTTP a livello globale utilizzando la seguente riga di codice.

```
s3 = Aws::S3::Client.new(:logger => Logger.new($stdout), :http_wire_trace => true)
```

Per suggerimenti su come ottenere informazioni sui bonifici da un AWS cliente, vedi Suggerimento per il debug: Ottenere informazioni sulla tracciabilità telefonica da un cliente.

Utilizzo di per AWS CLI ottenere gli ID delle richieste

Per ottenere gli ID della richiesta quando usi il AWS Command Line Interface (AWS CLI), aggiungi `--debug` al tuo comando.

Utilizzo di Windows PowerShell per ottenere gli ID delle richieste

Per informazioni sul recupero dei log con Windows PowerShell, consultate il post di blog Response Logging AWS Tools for Windows PowerShell in.NET Development.

Utilizzo degli eventi AWS CloudTrail relativi ai dati per ottenere gli ID delle richieste

Un bucket Amazon S3 configurato con eventi di CloudTrail dati per registrare le operazioni API a livello di oggetto S3 fornisce informazioni dettagliate sulle azioni intraprese da un utente, un ruolo o un servizio in Amazon S3. AWS Puoi identificare gli ID delle richieste S3 interrogando CloudTrail gli eventi con Athena.

Utilizzo della registrazione degli accessi al server S3 per recuperare gli ID richiesta

Un bucket Amazon S3 configurato per la registrazione degli accessi al server S3 fornisce record dettagliati per le richieste effettuate a tale bucket. È possibile identificare gli ID delle richieste S3 eseguendo query sui log degli accessi al server utilizzando Athena.

Cronologia dei documenti

- Versione API corrente: 2006-03-01

Nella tabella seguente vengono descritte le modifiche importanti apportate a ciascuna versione della Documentazione di riferimento delle API di Amazon Simple Storage Service e della Guida per l'utente di Amazon S3. Per ricevere notifiche sugli aggiornamenti di questa documentazione, puoi abbonarti a un feed RSS.

Modifica	Descrizione	Data
Amazon S3 Inventory supporta s3: condition key Inventory AccessibleOptionalFields	Amazon S3 Inventory supporta la chiave di InventoryAccessibleOptionalFields condizione s3: per controllare se gli utenti possono includere campi di metadati opzionali nei loro report. Per ulteriori informazioni, consulta la creazione della configurazione dei report di Control S3 Inventory.	20 febbraio 2024
Supporto IPv6 per S3 su Outposts	Ora puoi accedere ai bucket S3 on Outposts utilizzando IPv6 tramite S3 sugli endpoint dual-stack Outposts. Il supporto IPv6 per S3 on Outposts consente di gestire i bucket S3 on Outposts e controllare le risorse del piano di controllo sulle reti IPv6.	16 gennaio 2024
Nuova classe di archiviazione Amazon S3 a zona singola ad alte prestazioni: S3 Express One Zone	Amazon S3 Express One Zone è una classe di archiviazione Amazon S3 a zona singola ad alte prestazioni,	28 novembre 2023

	creata appositamente per fornire un accesso ai dati coerente di pochi millisecondi per le applicazioni sensibili alla latenza. Per ulteriori informazioni, consulta S3 Express One Zone.	
Mountpoint per Amazon S3 aggiunge il supporto per S3 Express One Zone	Ora puoi montare i bucket di directory S3 Express One Zone con Mountpoint.	28 novembre 2023
Versione dello schema di invocazione Lambda	Operazioni in batch Amazon S3 introduce una nuova versione dello schema di invocazione Lambda per l'utilizzo con processi Operazioni in batch che agiscono sui bucket di directory. Per ulteriori informazioni, consulta Utilizzo di Lambda e Operazioni in batch Amazon S3 con bucket di directory.	28 novembre 2023
Azione di importazione per bucket di directory	Amazon S3 introduce l'azione di importazione. L'importazione è un metodo ottimizzato di creazione di processi Operazioni in batch Amazon S3 per copiare oggetti da bucket per uso generico in bucket di directory. Per ulteriori informazioni, consulta Importing objects into a directory bucket.	28 novembre 2023

Gestione dell'accesso a S3 con S3 Access Grants	Amazon S3 Access Grants consente di gestire le autorizzazioni dei dati su larga scala per i principal i AWS Identity and Access Management (IAM) oltre alle identità di directory presenti nelle directory aziendali come. Azure AD Ora puoi applicare le autorizzazioni S3 con privilegio minimo e dimensionare facilment e tali autorizzazioni in base alle esigenze aziendali. Per ulteriori informazioni, consulta Managing access with S3 Access Grants.	26 novembre 2023
Mountpoint per Amazon S3 aggiunge funzionalità di memorizzazione nella cache	Con Mountpoint, ora puoi configurare la memorizza zione nella cache per i dati con accesso ripetuto.	22 novembre 2023

Generazione avanzata del manifesto di Amazon S3 Batch Operations	Ora puoi configurare Operazioni in batch Amazon S3 per generare automatic amente un manifesto in base ai criteri di filtro degli oggetti specificati quando si crea il processo. Questa opzione è disponibile per i lavori di replica in batch creati nella console Amazon S3 o per qualsiasi tipo di lavoro creato utilizzando gli SDK o AWS CLI l'API AWS REST di Amazon S3. Per ulteriori informazi oni, consulta Creazione di un processo di operazioni in batch S3.	22 novembre 2023
I bucket Amazon S3 esistenti possono ora aggiungere configurazioni Object Lock	Ora puoi abilitare Object Lock su un bucket S3 esistente. Puoi impostare blocchi a fini legali e periodi di conservaz ione per bucket nuovi o esistenti. Per ulteriori informazi oni, consulta Utilizzo del blocco oggetti S3.	20 novembre 2023
Parametri delle richieste di S3 Storage Lens per i prefissi	S3 Storage Lens introduce parametri delle richieste per i prefissi all'interno di un bucket Amazon S3. Per ulteriori informazioni, consulta Categorie di parametri.	17 novembre 2023

Gruppi Amazon S3 Storage Lens	S3 Storage Lens introduce i gruppi Storage Lens, un filtro definito su misura per gli oggetti in base ai metadati di un oggetto. Per maggiori informazioni, consulta Utilizzo di Amazon S3 Storage Lens con i gruppi.	15 novembre 2023
Nuova policy IAM	S3 su Outposts presenta `AWSServiceRoleForS3OnOutposts`, un ruolo collegato ai servizi per aiutarti a gestire le risorse di rete. Per ulteriori informazioni, consulta Using service-linked roles for Amazon S3 on Outposts.	3 ottobre 2023
Amazon S3 fornisce l'orario `Last-Modified` per i contrassegni di eliminazione	Amazon S3 fornisce l'orario `Last-Modified` per i contrassegni di eliminazione nelle intestazioni di risposta di S3 `Head` e delle operazioni API `Get`. Per ulteriori informazioni, consulta Utilizzo dei contrassegni di eliminazione.	27 settembre 2023
Aggiornamento Amazon S3 alla AWS policy gestita	Amazon S3 ha aggiunto le autorizzazioni `s3:Describe*` a `AmazonS3ReadOnlyAccess`. Per ulteriori informazioni, consulta Policy gestite da AWS per Amazon S3.	11 agosto 2023

Tempi di avvio migliorati per le richieste di ripristino Standard effettuate tramite Operazioni in batch Amazon S3	I recuperi standard per le richieste di ripristino effettuate tramite Operazioni in batch Amazon S3 ora possono iniziare in pochi minuti. Per ulteriori informazioni consulta Opzioni di recupero dall'archivio.	9 agosto 2023
È stato aggiunto Mountpoint, un client ad alta velocità di trasmissione effettiva per il montaggio di un bucket Amazon S3 come file system locale.	Con Mountpoint, le applicazioni possono accedere agli oggetti archiviati in Amazon S3 tramite operazioni sui file, dando alle applicazioni l'accesso all'archiviazione elastica e alla velocità di trasmissione effettiva di Amazon S3 tramite un'interfaccia di file.	9 agosto 2023
Crittografia lato server a doppio livello con chiavi (DSSE-KMS) AWS Key Management Service	La crittografia lato server a doppio livello con chiavi AWS Key Management Service (AWS KMS) (DSSE-KMS) applica due livelli di crittografia agli oggetti quando vengono caricati su Amazon S3. Per ulteriori informazioni, consulta Utilizzo della crittografia lato server a doppio livello con chiavi. AWS KMS	13 giugno 2023

Amazon S3 abilita il blocco dell'accesso pubblico S3 e disabilita le liste di controllo degli accessi (ACL) di S3 per tutti i nuovi bucket.	Amazon S3 ora abilita automaticamente S3 Block Public Access e disabilit a gli elenchi di controllo degli accessi (ACL) S3 per tutti i nuovi bucket S3 in tutte le regioni. AWS Per ulteriori informazioni, consulta Blocco dell'accesso pubblico allo storage Amazon S3 e Controllo della proprietà degli oggetti e disabilitazione delle ACL per il bucket.	27 aprile 2023
Metrica delle operazioni di Replica Amazon S3 non riuscite	Amazon S3 aggiunge una nuova Amazon CloudWatc h metrica per monitorare gli errori di replica S3. Per ulteriori informazioni, consulta Monitoraggio dell'avan zamento con i parametri di replica.	5 aprile 2023
DNS privato	AWS PrivateLink per Amazon S3 ora supporta il DNS privato. Per ulteriori informazi oni, consultare DNS privato.	14 marzo 2023
Supporto multi-account dei punti di accesso nella console Amazon S3	Amazon S3 ora supporta la creazione di punti di accesso multi-account nella propria console. Per ulteriori informazi oni, consulta Creazione dei punti di accesso.	14 marzo 2023

Amazon S3 su Outposts supporta Replica Amazon S3 su Outposts	Con la replica S3 locale, puoi replicare automaticamente gli oggetti in un singolo bucket Outposts di destinazione o in più bucket di destinazione. I bucket di destinazione possono trovarsi in Outposts diversi AWS Outposts o all'interno degli stessi Outposts del bucket di origine. Per ulteriori informazioni, consulta la pagina relativa alla replica di oggetti per S3 su Outposts.	14 marzo 2023
Alias del punto di accesso Lambda per oggetti Amazon S3	Quando crei un punto di accesso Lambda per oggetti, Amazon S3 genera automaticamente un alias univoco per il tuo punto di accesso Lambda per oggetti. Puoi utilizzare questo alias del punto di accesso al posto di un nome del bucket Amazon S3 o del nome della risorsa Amazon (ARN) del punto di accesso Lambda per oggetti in una richiesta per qualsiasi operazione del piano dati del punto di accesso. Per ulteriori informazioni, consulta la pagina relativa a come utilizzare un alias in stile bucket per il punto di accesso Lambda per oggetti.	14 marzo 2023

Supporto multi-account dei punti di accesso multi-regione in Amazon S3	Amazon S3 ora supporta la creazione di punti di accesso multi-regione multi-account con la console Amazon S3. Per ulteriori informazioni, consulta Creazione di punti di accesso multi-regione.	14 marzo 2023
Punti di accesso multi-account	Amazon S3 supporta la creazione di punti di accesso multi-account. È possibile creare un punto di accesso multi-account utilizzando la AWS Command Line Interface (AWS CLI) o l'operazione `CreateAccessPoint` della REST API. Per ulteriori informazioni, consulta Creazione dei punti di accesso.	30 novembre 2022

Amazon S3 supporta i controlli di failover per i punti di accesso multi-regione in Amazon S3	In Amazon S3 è stato introdotto il controllo di failover per i punti di accesso multi-regione. Questi controlli consentono di spostare il traffico delle richieste di accesso ai dati S3 indirizzato da un punto di accesso multi-regione Amazon S3 a una Regione AWS alternativa in pochi minuti per testare e creare applicazioni a disponibilità elevata. Per ulteriori informazioni, consulta la sezione relativa ai controlli di failover dei punti di accesso multi-regione Amazon S3.	28 novembre 2022
Amazon S3 Storage Lens aumenta la visibilità a livello di organizzazione con 34 nuovi parametri	S3 Storage Lens introduce altri 34 parametri per individuare migliori opportunità di ottimizzazione dei costi, identificare le best practice per la protezione dei dati e migliorare le prestazioni dei flussi di lavoro delle applicazioni. Per ulteriori informazioni, consulta la sezione Parametri di S3 Storage Lens.	17 novembre 2022

Amazon S3 supporta richieste di ripristino a velocità più elevate per le classi di archiviazione S3 Glacier Flexible Retrieval (Recupero flessibile S3 Glacier) e S3 Glacier Deep Archive (Archiviazione profonda S3 Glacier)	Amazon S3 supporta le richieste di ripristino a una velocità massima di 1.000 transazioni al secondo, Account AWS per le classi di storage S3 Glacier Flexible Retrieval e S3 Glacier Deep Archive.	15 novembre 2022
Amazon S3 su Outposts supporta azioni e filtri del ciclo di vita S3 aggiuntivi	S3 su Outposts supporta regole aggiuntive del ciclo di vita S3 per ottimizzare la gestione della capacità. È possibile far scadere gli oggetti man mano che invecchiano o vengono sostituiti da versioni più recenti. È possibile creare una regola del ciclo di vita per un intero bucket o un sottoinsieme di oggetti in un bucket filtrando per prefissi, tag di oggetto o dimensione di oggetto. Per ulteriori informazioni, consulta Creazione e gestione di una configurazione del ciclo di vita.	2 novembre 2022
Supporto della replica S3 per oggetti SSE-C	È possibile replicare gli oggetti creati con crittografia lato server mediante chiavi fornite dal cliente. Per ulteriori informazioni sulla replica di oggetti crittografati, consulta Replica di oggetti creati con crittografia lato server (SSE-C, SSE-S3, SSE-KMS).	24 ottobre 2022

Amazon S3 su Outposts supporta gli alias del punto di accesso	Con S3 su Outposts devi utilizzare i punti di accesso per accedere a qualsiasi oggetto in un bucket Outposts. Ogni volta che crei un punto di accesso per un bucket, S3 su Outposts genera automatic amente un alias per tale punto di accesso. Puoi utilizzar e questo alias del punto di accesso al posto dell'ARN del punto di accesso per qualsiasi operazione del piano dati. Per ulteriori informazioni, consulta Using a bucket-style alias for your S3 on Outposts bucket access point (Utilizzo di un alias in stile bucket per il punto di accesso del bucket S3 su Outposts).	21 ottobre 2022
S3 Object Lambda supporta le operazioni `HeadObject` , `ListObjects` e `ListObjectsV2`	Puoi utilizzare il codice personalizzato per modificar e i dati restituiti dalle richieste S3 standard `GET`, `LIST` o `HEAD` per filtrare le righe, ridimensionare le immagini in modo dinamico, oscurare i dati riservati e molto altro. Per ulteriori informazioni, consulta Trasformazione di oggetti con S3 Object Lambda.	4 ottobre 2022

Amazon S3 su Outposts supporta il controllo delle versioni S3	Se abilitato, il controllo delle versioni S3 conserva più copie distinte di un oggetto nello stesso bucket. Puoi utilizzare il controllo delle versioni S3 per conservare, recuperare e ripristinare qualsiasi versione di ogni oggetto archiviato nei bucket Outposts. Il controllo delle versioni S3 ti consente di eseguire il ripristino a seguito di errori dell'applicazione e operazioni non intenzionali degli utenti. Per ulteriori informazioni, consulta Managing S3 Versioning for your S3 on Outposts bucket (Gestione del controllo delle versioni S3 per un bucket S3 su Outposts).	21 settembre 2022
AWS Backup per Amazon S3	AWS Backup è un servizio completamente gestito e basato su policy che puoi utilizzare per definire una policy di backup centrale per proteggere i tuoi dati Amazon S3. Per ulteriori informazioni, consulta Using AWS Backup for Amazon S3.	18 febbraio 2022

Utilizzo di S3 Batch Replication per replicare gli oggetti esistenti	S3 Batch Replication ti consente di replicare gli oggetti che esistevano già prima della configurazione della replica. La replica degli oggetti esistenti avviene tramite l'uso di un processo di operazioni in batch. S3 Batch Replication differisce dalla replica in tempo reale, che copia in modo continuo e automatico nuovi oggetti tra bucket Amazon S3. Per ulteriori informazioni, consulta la sezione Replica di oggetti esistenti con S3 Batch Replication.	8 febbraio 2022
Rinominazione di S3 Glacier Flexible Retrieval	La classe di archiviazione Glacier è stata rinominata S3 Glacier Flexible Retrieval. Questa modifica non ha alcun impatto sull'API.	30 novembre 2021

Nuova impostazione di S3 Object Ownership per disabilitare le ACL	Puoi applicare l'impostazione del proprietario del bucket per Object Ownership (Proprietà dell'oggetto) per disabilitare le liste di controllo degli accessi per il tuo bucket e gli oggetti contenuti e assumere la proprietà di ogni oggetto del tuo bucket. L'impostazione applicata del proprietario del bucket semplifica la gestione degli accessi per i dati archiviati in Amazon S3. Per ulteriori informazioni, consulta Controllo della proprietà degli oggetti e disabilitazione delle ACL per il bucket.	30 novembre 2021
Nuova classe di archiviazione S3 Intelligent-Tiering	S3 Intelligent-Tiering Archive Instant Access è una classe di archiviazione aggiuntiva di S3 Intelligent-Tiering. Per ulteriori informazioni, consulta Come funziona S3 Intelligent-Tiering.	30 novembre 2021
Nuova classe di archiviazione S3 Glacier Instant Retrieval	Ora è possibile posizionare gli oggetti nella classe di archiviazione S3 Glacier Instant Retrieval. Per maggiori informazioni su questa classe di archiviazione, consulta Utilizzare le classi di archiviazione di Amazon S3.	30 novembre 2021

AWS Backup per Amazon S3 Preview	AWS Backup è un servizio completamente gestito e basato su policy che puoi utilizzare per definire una policy di backup centrale per proteggere i tuoi dati Amazon S3. Per ulteriori informazioni, consulta Using AWS Backup for Amazon S3.	30 novembre 2021
AWS Identity and Access Management Access Analyzer per Amazon S3	IAM Access Analyzer esegue controlli della policy per convalidarla in rapporto alla sintassi della policy e alle best practice di IAM. Per ulteriori informazioni sulla convalida delle policy con IAM Access Analyzer, consulta Convalida delle policy di IAM Access Analyzer nella Guida per l'utente di IAM.	30 novembre 2021
Nuovi tipi di eventi	Nuovi tipi di eventi aggiunti alle notifiche di eventi di Amazon S3, consulta Notifiche di eventi Amazon S3.	29 novembre 2021
Abilita Amazon EventBridge sui bucket	Puoi abilitare i bucket EventBridge Amazon S3 per inviare eventi ad Amazon EventBridge, vedi Utilizzo. EventBridge	29 novembre 2021

Nuovi filtri del Ciclo di vita S3	È possibile creare regole del ciclo di vita in base alle dimensioni dell'oggetto o specificare il numero di versioni non correnti dell'oggetto da conservare. Per ulteriori informazioni, consulta Esempi di configurazione del ciclo di vita S3.	23 novembre 2021
Pubblica i parametri di Amazon S3 Storage Lens su Amazon CloudWatch	Puoi pubblicare i parametri di utilizzo e attività di S3 Storage Lens su Amazon CloudWatch per creare una visione unificata dello stato di salute operativo nei dashboard. CloudWatch Puoi anche utilizzare CloudWatch funzionalità, come allarmi e azioni attivate, calcoli metrici e rilevamento delle anomalie, per monitorare e agire in base ai parametri di S3 Storage Lens. Inoltre, le CloudWatch API consentono alle applicazioni, inclusi i provider di terze parti, di accedere alle metriche di S3 Storage Lens. Per ulteriori informazioni, consulta le metriche di Monitor S3 Storage Lens in. CloudWatch	22 novembre 2021

Punti di accesso multi-regione	Puoi utilizzare i punti di accesso multi-regione per creare un endpoint globale utilizzabile dalle applicazioni per eseguire le richieste provenienti da bucket Amazon S3 situati in più Regioni AWS. Puoi utilizzare questo punto di accesso multi-regione per instradare i dati al bucket con la latenza più bassa. Per ulteriori informazioni sui punti di accesso multi-regione e su come utilizzarli, consulta Punto di accesso multi-regione in Amazon S3.	2 settembre 2021
Amazon S3 su Outposts aggiunge l'accesso locale diretto per le applicazioni	Esegui le tue applicazioni al di fuori del cloud privato AWS Outposts virtuale (VPC) e accedi ai dati di S3 on Outposts. È inoltre possibile accedere agli oggetti S3 su Outposts direttamente dalla rete on-premise. Per ulteriori informazioni sulla configurazione di S3 su Outposts tramite Indirizzi IP di proprietà del cliente(CoIP) e accedendo ai tuoi oggetti creando un gateway locale dalla rete on-premise, consultaAccesso ad Amazon S3 su Outposts solo tramite punti di accesso del Virtual Private Cloud (VPC).	29 luglio 2021

Alias del punto di accesso Amazon S3	Quando crei un punto di accesso, Amazon S3 genera automaticamente un alias che puoi utilizzare al posto del nome del bucket per l'accesso ai dati. Puoi utilizzare questo alias del punto di accesso al posto di un Amazon Resource Name (ARN) per qualsiasi operazione del piano dati del punto di accesso. Per ulteriori informazioni, consulta Utilizzo di un alias in stile bucket per il punto di accesso.	26 luglio 2021
Amazon S3 Inventory e Operazioni in batch Amazon S3 supportano lo stato della chiave del bucket S3	Le operazioni di inventario e in batch di Amazon S3 supportano l'identificazione e la copia di oggetti esistenti con le chiavi bucket S3. Le chiavi bucket S3 accelerano la riduzione dei costi di crittografia lato server per gli oggetti esistenti. Per ulteriori informazioni, consulta Inventario Amazon S3e Oggetto Copia per operazioni in batch.	3 giugno 2021

Snapshot dell'account con i parametri di Amazon S3 Storage Lens	Lo snapshot dell'account S3 Storage Lens mostra l'archiviazione totale, il numero di oggetti e la dimension e media degli oggetti nella pagina Home della console S3 (Bucket) riepilogando i parametri della dashboard predefinita. Per ulteriori informazioni, consultare lo snapshot dell'account dei parametri di S3 Storage Lens.	5 maggio 2021
Aumento del supporto per endpoint Amazon S3 su Outposts	S3 su Outposts supporta ora fino a 100 endpoint per Outpost. Per ulteriori informazi oni, vedere Limitazioni di rete di S3 su Outposts.	29 aprile 2021
Amazon S3 on Outposts: notifiche di eventi in Amazon Events CloudWatch	Puoi utilizzare CloudWatch Events per creare una regola per acquisire qualsiasi evento dell'API S3 on Outposts e ricevere notifiche tramite tutte le CloudWatch destinazi oni supportate. Per ulteriori informazioni, consulta Ricezione delle notifiche degli eventi di S3 on Outposts tramite CloudWatch Events.	19 aprile 2021

S3 Object Lambda	Con S3 Object Lambda puoi aggiungere il tuo codice alle richieste GET di Amazon S3 per modificare ed elaborare i dati quando vengono restituiti a un'applicazione. Puoi utilizzare il codice personalizzato per modificare i dati restituiti dalle richieste GET S3 standard per filtrare le righe, ridimensionare le immagini in modo dinamico, oscurare i dati riservati e molto altro. Per ulteriori informazioni, consulta Trasformazione di oggetti.	18 marzo 2021
AWS PrivateLink	Con AWS PrivateLink per Amazon S3, puoi connetterti direttamente a S3 utilizzando un endpoint di interfaccia nel tuo cloud privato virtuale (VPC) anziché collegarti tramite Internet. Gli endpoint di interfaccia sono direttamente accessibili dalle applicazioni che si trovano on-premise o in una Regione AWS diversa. Per ulteriori dettagli, consulta AWS PrivateLink per Amazon S3.	2 febbraio 2021

Gestione della capacità di Amazon S3 on Outposts con AWS CloudTrail	Gli eventi di gestione di S3 on Outposts sono disponibili CloudTrail tramite i log. Per ulteriori informazioni, consulta Gestire la capacità di S3 on Outposts con. CloudTrail	21 dicembre 2020
Forte coerenza	Amazon S3 offre una forte read-after-write coerenza PUT e in generale DELETE le richieste di oggetti nel bucket S3. Regioni AWS Inoltre, le operazioni di lettura su Amazon S3 Select, le liste di controllo degli accessi Amazon S3, i tag di oggetti Amazon S3 e i metadati di oggetti (ad esempio l'oggetto HEAD) sono fortemente consisten ti. Per maggiori informazi oni, consulta il modello di consistenza dei dati di Amazon S3.	1 dicembre 2020

Sincronizzazione delle modifiche alla replica di Amazon S3	La sincronizzazione delle modifiche alla replica di Amazon S3 mantiene sincronizzati i metadati degli oggetti come tag, ACL e le impostazioni di blocco oggetti tra gli oggetti di origine e le repliche. Quando questa funzione è abilitata, Amazon S3 replica le modifiche ai metadati apportate all'oggetto di origine o alle copie di replica. Per ulteriori informazioni, consulta Replicating metadata changes with replica modification sync (Replica delle modifiche ai metadati con sincronizzazione delle modifiche alla replica).	1 dicembre 2020
Chiavi bucket Amazon S3	Le chiavi del bucket Amazon S3 riducono il costo della crittografia lato server di Amazon S3 con AWS Key Management Service (SSE-KMS). Questa nuova chiave a livello di bucket per la crittografia lato server può diminuire i costi delle richieste di AWS KMS fino al 99% riducendo il traffico delle richieste da Amazon S3 a AWS KMS. Per ulteriori informazioni, consulta Riduzione del costo di SSE-KMS con le chiavi bucket Amazon S3.	1 dicembre 2020

Amazon S3 Storage Lens	S3 Storage Lens aggrega i tuoi parametri e mostra le informazioni nella sezione Account snapshot (Snapshot dell'account) nella pagina Buckets (Bucket) della console Amazon S3. S3 Storage Lens dispone inoltre di un pannello di controllo interattivo che puoi usare per visualizzare informazioni dettagliate e tendenze, contrassegnare le anomalie e ricevere consigli per ottimizzare i costi di archiviazione e applicare le best practice sulla protezione dei dati. Nel pannello di controllo sono disponibili opzioni di drill-down per generare e visualizzare approfondimenti a livello di organizzazione, account, Regione AWS, classe di archiviazione, bucket, prefisso o gruppo Storage Lens. Puoi anche inviare un'esportazione dei parametri quotidiana in formato CSV o Parquet a un bucket S3. Per ulteriori informazioni, consulta Valutazione dell'attività e dell'utilizzo dello storage con S3 Storage Lens.	18 novembre 2020

Tracciamento delle richieste S3 tramite AWS X-Ray	Amazon S3 si integra con X-Ray per propagare il contesto di traccia e fornire una catena di richieste con nodi a monte e a valle. Per ulteriori informazioni, consulta Tracciamento delle richieste tramite X-Ray.	16 Novembre 2020
Parametri di replica S3	I parametri di replica S3 forniscono parametri dettagliati per le regole nella configurazione di replica. Per maggiori informazioni, consulta Replication metrics and Amazon S3 event notifications (Parametri di replica e notifiche di eventi Amazon S3).	9 novembre 2020
Accesso Intelligent-Tiering Archive e accesso Deep Archive di S3	L'accesso S3 Intelligent-Tiering Archive e l'accesso Deep Archive sono livelli di storage aggiuntivi di S3 Intelligent-Tiering. Per ulteriori informazioni, consulta La classe di storage che ottimizza automaticamente gli oggetti con accesso più o meno frequente.	9 novembre 2020

Replica del contrassegno di eliminazione	Con la replica del contrassegno di eliminazione è possibile garantire che i contrassegni di eliminazione vengano copiati nei bucket di destinazione per le regole di replica. Per ulteriori informazioni, consulta Utilizzo della replica dei contrassegni di eliminazione.	9 novembre 2020
S3 Object Ownership	Object Ownership è un'impostazione del bucket S3 che è possibile utilizzare per controllare la proprietà dei nuovi oggetti che vengono caricati nei bucket. Per ulteriori informazioni, consulta Utilizzo di S3 Object Ownership.	2 ottobre 2020
Amazon S3 su Outposts	Con Amazon S3 on Outposts, puoi creare bucket S3 sulle tue AWS Outposts risorse e archiviare e recuperare facilmente oggetti in locale per applicazioni che richiedono l'accesso locale ai dati, l'elaborazione locale dei dati e la residenza dei dati. Puoi usare S3 su Outposts tramite AWS Management Console AWS CLI, AWS , SDK o API REST. Per ulteriori informazioni, consulta Utilizzo di Amazon S3 su Outposts.	30 settembre 2020

Condizione proprietario del bucket	Puoi utilizzare la condizione di proprietario del bucket Amazon S3 per assicurarti che i bucket utilizzati nelle tue operazioni S3 appartengano a quelli che ti aspetti. Account AWS Per ulteriori informazioni, consulta Condizione del proprietario del bucket.	11 settembre 2020
Supporto delle operazioni in batch S3 per la conservazione del blocco oggetti	È ora possibile utilizzare le operazioni in batch con il blocco oggetti S3 per applicare le impostazioni di conservazione a molti oggetti Amazon S3 contemporaneamente. Per ulteriori informazioni, consulta Setting S3 Object Lock Retention dates with S3 Batch Operations (Gestione delle date di conservazione del blocco oggetti S3 con Operazioni di batch S3).	4 maggio 2020
Supporto delle operazioni in batch S3 per il blocco di carattere legale del blocco oggetti	Puoi utilizzare Operazioni di batch con il blocco oggetti S3 per aggiungere blocchi di carattere legale a molti oggetti Amazon S3 contemporaneamente. Per ulteriori informazioni, consulta Utilizzo di S3 Batch Operations per impostare il blocco di carattere legale del blocco oggetti S3.	4 maggio 2020

Tag dei processi per Operazioni di batch S3	È possibile aggiungere tag ai processi di operazioni in batch Amazon S3 per controllare ed etichettare tali processi. Per ulteriori informazioni, consulta Tags for S3 Batch Operations jobs (Tag dei processi per Operazioni di batch S3).	16 marzo 2020
Access point Amazon S3	I punti di accesso Amazon S3 semplificano la gestione dell'accesso ai dati su vasta scala per set di dati condivisi in S3. Gli access point sono endpoint di rete denominati che vengono collegati a bucket che puoi usare per eseguire operazioni su oggetti S3. Per ulteriori informazioni, consulta Gestione dell'accesso ai dati con access point Amazon S3.	2 dicembre 2019
Access Analyzer per Amazon S3	Access Analyzer per Amazon S3 ti avvisa dei bucket S3 configurati per consentire l'accesso a chiunque su Internet o Account AWS altro, compresi gli account esterni alla tua organizzazione. Per ulteriori informazioni, consulta Utilizzo di Access Analyzer per Amazon S3.	2 dicembre 2019

S3 Replication Time Control (S3 RTC)	S3 Replication Time Control (S3 RTC) replica la maggior parte degli oggetti caricati in Amazon S3 in pochi secondi, arrivando al 99,99% di tali oggetti in 15 minuti. Per ulteriori informazioni, consulta Replicating objects using S3 Replication Time Control (S3 RTC) (Replica di oggetti utilizzando il controllo del tempo di replica di S3 (S3 RTC)).	20 novembre 2019
Replica nella stessa regione	La replica nella stessa Regione (Same-Region Replication, SRR) viene utilizzata per copiare gli oggetti tra bucket Amazon S3 nella stessa Regione AWS. Per informazioni sulla replica tra Regioni e nella stessa Regione, consulta Replica di oggetti.	18 settembre 2019
Supporto della replica tra regioni per il blocco oggetti S3	La replica tra regioni supporta ora il blocco oggetti. Per ulteriori informazioni, consulta Cosa replica Amazon S3?.	28 maggio 2019

Operazioni in batch S3	Utilizzando Operazioni di batch S3 è possibile eseguire le operazioni in batch su vasta scala su oggetti Amazon S3. Le operazioni in batch S3 possono eseguire una singola operazione su elenchi di oggetti specificati. Un solo processo può eseguire l'operazione specificata su miliardi di oggetti contenent i exabyte di dati. Per ulteriori informazioni, consulta Esecuzione di S3 Batch Operations.	30 Aprile 2019
Regione Asia Pacifico (Hong Kong)	Amazon S3 è ora disponibi le nella regione Asia Pacifico (Hong Kong). Per ulteriori informazioni sulle regioni e gli endpoint Amazon S3, consultare la sezione relativa a regioni ed endpoint nella Riferimenti generali di AWS.	24 aprile 2019
Aggiunto un nuovo campo ai log di accesso al server	In Amazon S3 è stato aggiunto il seguente nuovo campo ai log di accesso al server: Transport Layer Security (TLS) version (Versione di Transport Layer Security (TLS)). Per ulteriori informazioni, consulta Formato del log di accesso al server.	28 marzo 2019

Nuova classe di storage per l'archiviazione	Amazon S3 offre ora una nuova classe di archiviazione per gli oggetti con accesso non frequente: Deep Archive S3 Glacier (`DEEP_ARCHIVE`). Per ulteriori informazioni, consulta Classi di storage.	27 marzo 2019
Aggiunti nuovi campi ai log di accesso al server	In Amazon S3 sono stati aggiunti i seguenti nuovi campi ai log di accesso al server: Host Id (ID host), Signature Version (Versione firma), Cipher Suite (Pacchetto di crittografia), Authentication Type (Tipo di autenticazione) e Host Header (Intestazione host). Per ulteriori informazioni, consulta Formato del log di accesso al server.	5 marzo 2019
Supporto per i file di Inventario Amazon S3 in formato Parquet	Amazon S3 supporta ora il formato Apache Parquet (Parquet) in aggiunta ai formati di file Apache ORC (Optimized Row Columnar) e CSV (Comma-Separated Values, valori separati da virgole) per i file di output di inventario. Per ulteriori informazioni, consulta Inventario.	4 dicembre 2018

Blocco di oggetti in S3	Amazon S3 offre ora una funzionalità di blocco oggetti che fornisce la protezione WORM (Write Once Read Many) per gli oggetti Amazon S3. Per ulteriori informazioni, consulta Blocco degli oggetti.	26 Novembre 2018
Aggiornamento della velocità di ripristino	Tramite l'aggiornamento della velocità di ripristino in Amazon S3 è possibile modificare la velocità di un'operazione di ripristino dalla classe di archiviazione S3 Glacier Flexible Retrieval (Recupero flessibile S3 Glacier) aumentandola mentre il ripristino è in corso. Per ulteriori informazioni, consulta Ripristino di oggetti archiviati.	26 Novembre 2018
Notifiche di eventi di ripristino	La funzionalità di notifica eventi di Amazon S3 ora supporta eventi di avvio e completamento durante il ripristino di oggetti dalla classe di archiviazione S3 Glacier Flexible Retrieval (Recupero flessibile S3 Glacier). Per ulteriori informazioni, consulta Notifiche di eventi.	26 novembre 2018

PUT direttamente nella classe di archiviazione recupero flessibile S3 Glacier	L'operazione PUT in Amazon S3 permette ora di specificare il recupero flessibile S3 Glacier come classe di archiviazione quando si creano oggetti. Nelle versioni precedenti era necessario eseguire la transizione degli oggetti nella classe di archiviazione S3 Glacier Flexible Retrieval da un'altra classe di archiviazione Amazon S3. Adesso inoltre, quando viene utilizzato S3 Cross-Region Replication (CRR), è possibile specificare S3 Glacier Flexible Retrieval come classe di archiviazione per gli oggetti replicati. Per ulteriori informazioni sulla classe di archiviazione S3 Glacier Flexible Retrieval, consulta Classi di archiviazione. Per ulteriori informazioni su come specificare la classe di storage per gli oggetti replicati, consulta Panoramica della configurazione di replica. Per ulteriori informazioni sulle modifiche apportate dall'operazione PUT diretta alla REST API nel recupero flessibile S3 Glacier, consulta Document History: PUT directly to S3 Glacier Flexible Retrieval (Cronologia dei documenti:	26 novembre 2018

	PUT direttamente nel recupero flessibile S3 Glacier).	
Nuova classe di storage	Amazon S3 offre ora una nuova classe di archiviazione denominata S3 Intelligent-Tiering (`INTELLIGENT_TIERING`) progettata per dati di lunga durata con modelli di accesso variabili o sconosciuti. Per ulteriori informazioni, consulta Classi di storage.	26 Novembre 2018
Blocco dell'accesso pubblico di Amazon S3	Amazon S3 permette ora di bloccare l'accesso pubblico a bucket e oggetti a livello di singolo bucket o di account. Per ulteriori informazioni, consulta Utilizzo del blocco dell'accesso pubblico di Amazon S3.	15 Novembre 2018

Miglioramenti ai filtri nelle regole di replica tra regioni	Nella configurazione di una regola di replica tra regioni puoi specificare un filtro di oggetti per scegliere un sottoinsieme di oggetti cui applicare la regola. Nelle versioni precedenti era possibile filtrare solo in base a un prefisso della chiave di un oggetto. In questa versione puoi filtrare in base al prefisso della chiave di un oggetto, a uno o più tag dell'oggetto o a entrambe le condizioni. Per ulteriori informazioni, consulta Configurazione di CRR: panoramica della configurazione di replica.	19 settembre 2018
Nuove caratteristiche di Amazon S3 Select	Amazon S3 Select ora supporta l'input di Apache Parquet, le query su oggetti JSON annidati e due nuove metriche di monitoraggio Amazon CloudWatch (and). `SelectScannedBytes` `SelectReturnedBytes`	5 settembre 2018
Aggiornamenti ora disponibili tramite RSS	È ora possibile abbonarsi a un feed RSS per ricevere notifiche sugli aggiornamenti alla Guida per l'utente di Amazon S3.	19 giugno 2018

Aggiornamenti precedenti

La tabella seguente descrive le modifiche importanti introdotte in ogni versione della Guida per l'utente di Amazon S3 prima del 19 giugno 2018.

Modifica	Descrizione	Data
Aggiornamento degli esempi di codice	Esempi di codice aggiornati: • C# – Aggiornamento di tutti gli esempi per l'utilizzo del modello asincrono basato su attività. Per ulteriori informazioni, consulta Amazon Web Services Asynchronous APIs for .NET nella Developer Guide. AWS SDK for .NET Gli esempi di codice sono ora conformi alla versione 3 di AWS SDK for .NET. • Java – Aggiornamento di tutti gli esempi per utilizzare il modello del generatore client. Per ulteriori informazioni sul modello del generatore client, consulta Creazione di client del servizio. • PHP: tutti gli esempi sono stati aggiornati per utilizzare AWS SDK for PHP 3.0. Per ulteriori informazioni sulla versione 3.0, consulta. AWS SDK for PHP AWS SDK for PHP • Ruby: codice di esempio aggiornato in modo che gli esempi funzionino con la AWS SDK for Ruby versione 3.	30 Aprile 2018
Amazon S3 ora riporta le classi di recupero flessibile e di storage di S3 Glacier ai parametri di `ONEZONE_IA` storage di Amazon Logs CloudWatch	Oltre a fare riferimento a byte effettivi, i parametri di archiviazione includono byte in sovraccarico per oggetto per le classi di archiviazione applicabili (`ONEZONE_IA` , `STANDARD_IA` e S3 Glacier Flexible Retrieval [Recupero flessibile S3 Glacier]): •	30 Aprile 2018

Modifica	Descrizione	Data
	Per gli oggetti delle classi di storage `ONEZONE_IA` e `STANDARD_IA`, Amazon S3 segnala gli oggetti con dimensioni inferiori a 128 KB come 128 KB. Per ulteriori informazioni, consulta Utilizzo delle classi di storage di Amazon S3. • Per gli oggetti della classe di archiviazione S3 Glacier Flexible Retrieval, i parametri di archiviazione segnalano i seguenti sovraccarichi: • Un sovraccarico di 32 KB per oggetto, addebitato in base ai prezzi della classe di archiviazione S3 Glacier Flexible Retrieval • Un sovraccarico di 8 KB per oggetto, addebitato secondo le tariffe della classe di archiviazione `STANDARD` Per ulteriori informazioni, consulta Trasferimento degli oggetti utilizzando il ciclo di vita Amazon S3. Per ulteriori informazioni sui parametri dello storage, consulta Monitoraggio delle metriche con Amazon CloudWatch.	
Nuova classe di storage	Amazon S3 offre ora una nuova classe di archiviazione, `STANDARD_IA` (IA sta per "Infrequent Access", accesso infrequente) per archiviare gli oggetti. Questa classe di storage è ottimizzata per i dati esistenti da molto tempo a cui si accede meno frequentemente. Per ulteriori informazioni, consulta Utilizzo delle classi di storage di Amazon S3.	4 Aprile 2018

Modifica	Descrizione	Data
Amazon S3 Select	Amazon S3 supporta ora il recupero del contenuto degli oggetti in base a un'espressione SQL. Per ulteriori informazioni, consulta Filtro e recupero dei dati tramite Amazon S3 Select.	4 Aprile 2018
Regione Asia Pacifico (Osaka-Locale)	Amazon S3 è ora disponibile nella regione Asia Pacifico (Osaka-Locale). Per ulteriori informazioni sulle regioni e gli endpoint Amazon S3, consultare la sezione relativa a regioni ed endpoint nella Riferimenti generali di AWS. **Important** È possibile utilizzare la regione Asia Pacifico (Osaka-Locale) solo in combinazione con la regione Asia Pacifico (Tokyo). Per richiedere l'accesso alla regione Asia Pacifico (Osaka-Lo cale), contatta il tuo rappresentante commerciale.	12 febbraio 2018
Timestamp di creazione di Amazon S3 Inventory	Amazon S3 Inventory include ora un timestamp che indica la data e l'ora di inizio della creazione del report di Inventari o Amazon S3. È possibile utilizzare il timestamp per determinare le modifiche nello storage Amazon S3 dall'ora di inizio della generazione del report di inventario.	16 gennaio 2018
Regione Europa (Parigi)	Amazon S3 è ora disponibile nella regione UE (Parigi). Per ulteriori informazioni sulle regioni e gli endpoint Amazon S3, consultare la sezione relativa a regioni ed endpoint nella Riferimenti generali di AWS.	18 dicembre 2017
Regione Cina (Ningxia)	Amazon S3 è ora disponibile nella regione Cina (Ningxia) . Per ulteriori informazioni sulle regioni e gli endpoint Amazon S3, consultare la sezione relativa a regioni ed endpoint nella Riferimenti generali di AWS.	29 Novembre 2017

Modifica	Descrizione	Data
Supporto per i file di Amazon S3 Inventory in formato ORC	Amazon S3 supporta ora il formato Apache ORC (Optimized Row Columnar) in aggiunta al formato di file CSV (Comma-Separated Values, valori separati da virgole) per i file di output di inventario. È inoltre possibile eseguire query sull'inventario Amazon S3 utilizzando SQL standard con Amazon Athena, Amazon Redshift Spectrum e altri strumenti, tra cui Presto, Apache Hive e Apache Spark. Per ulteriori informazioni, consulta Amazon S3 Inventory.	17 Novembre 2017
Crittografia predefinita per i bucket S3	La crittografia predefinita di Amazon S3 offre un modo per impostare il comportamento di crittografia predefinito di un bucket S3. Puoi configurare la crittografare predefinita di un bucket in modo che gli oggetti siano crittografati quando vengono memorizzati nel bucket. Gli oggetti vengono crittografati utilizzando la crittografia lato server con chiavi gestite Amazon S3 (SSE-S3) o chiavi gestite (SSE-KMS) AWS . Per ulteriori informazioni, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3.	06 Novembre 2017
Stato della crittografia in Amazon S3 Inventory	Amazon S3 supporta ora l'inserimento dello stato della crittografia in Amazon S3 Inventory per verificare come sono crittografati gli oggetti quando sono inattivi, per il controllo della conformità o per altri scopi. Inoltre, è possibile configurare la crittografia di Amazon S3 Inventory con crittografia lato server (SSE) o SSE-KMS in modo che tutti i file dell'inventario siano crittografati di conseguenza. Per ulteriori informazioni, consulta Amazon S3 Inventory.	06 Novembre 2017

Modifica	Descrizione	Data
Miglioramenti della replica tra regioni	La replica tra regioni ora supporta le seguenti caratteri stiche: • In uno scenario con più account, è possibile aggiunger e una configurazione CRR (Cross-Region Replication, replica tra regioni) per trasferire la proprietà della replica all' Account AWS proprietario del bucket di destinazione. Per ulteriori informazioni, consulta Modifica del proprieta rio della replica. • Per impostazione predefinita, Amazon S3 non replica gli oggetti nel bucket di origine creati utilizzando la crittogra fia lato server utilizzando le chiavi archiviate AWS KMS nella configurazione CRR, ora puoi ordinare ad Amazon S3 di replicare questi oggetti. Per ulteriori informazioni, consulta Replica di oggetti crittografati (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).	06 Novembre 2017
Europe (London) Region	Amazon S3 è ora disponibile nella regione UE (Londra). Per ulteriori informazioni sulle regioni e gli endpoint Amazon S3, consultare la sezione relativa a regioni ed endpoint nella Riferimenti generali di AWS.	13 dicembre 2016
Canada (Central) Region	Amazon S3 ora è disponibile nella regione Canada (Centrale). Per ulteriori informazioni sulle regioni e gli endpoint Amazon S3, consultare la sezione relativa a regioni ed endpoint nella Riferimenti generali di AWS.	8 dicembre 2016

Modifica	Descrizione	Data
Tagging oggetti	Amazon S3 supporta ora il tagging degli oggetti. Il tagging consente di catalogare lo storage. I prefissi dei nomi delle chiavi degli oggetti consentono di categorizzare lo storage, mentre il tagging aggiunge un'altra dimensione allo storage. Il tagging offre ulteriori benefici. Eccone alcuni: • I tag degli oggetti consentono un controllo estremamente preciso delle autorizzazioni (ad esempio, si potrebbero concedere a un utente IAM autorizzazioni di sola lettura sugli oggetti con tag specifici). • Controllo estremamente preciso della configurazione del ciclo di vita. È possibile specificare dei tag per seleziona re un sottoinsieme di oggetti a cui si applica la regola del ciclo di vita. • Se la replica tra regioni è configurata, Amazon S3 può replicare i tag. È necessario concedere l'autorizzazione appropriata al ruolo IAM creato affinché Amazon S3 si incarichi di replicare gli oggetti automaticamente. • Puoi anche personalizzare CloudWatch metriche ed CloudTrail eventi per visualizzare informazioni tramite filtri di tag specifici. Per ulteriori informazioni, consulta Suddivisione in categorie dello storage utilizzando i tag.	29 Novembre 2016

Modifica	Descrizione	Data
Il ciclo di vita di Amazon S3 supporta ora i filtri basati su tag	Amazon S3 supporta ora i filtri basati su tag nella configurazione del ciclo di vita. Ora si può specificare una regola del ciclo di vita del bucket in cui è possibile indicare un prefisso della chiave, uno o più tag dell'oggetto o una combinazione di questi elementi per selezionare un sottoinsieme di oggetti a cui si applica la regola del ciclo di vita. Per ulteriori informazioni, consulta Gestione del ciclo di vita dello storage.	29 Novembre 2016
CloudWatch richiedi le metriche per i bucket	Amazon S3 ora supporta i CloudWatch parametri per le richieste effettuate sui bucket. Quando vengono abilitati per un bucket, questi parametri sono disponibili a intervalli di 1 minuto. È inoltre possibile definire quali oggetti in un bucket riporteranno i parametri di richiesta. Per ulteriori informazioni, consulta Monitoraggio delle metriche con Amazon CloudWatch.	29 Novembre 2016
Inventario Amazon S3	Amazon S3 supporta ora l'inventario dello storage. Amazon S3 Inventory genera un file di output flat degli oggetti e dei metadati corrispondenti per un bucket S3 o un prefisso condiviso (ovvero oggetti con nomi che iniziano con una stringa comune), su base giornaliera o settimanale. Per ulteriori informazioni, consulta Amazon S3 Inventory.	29 Novembre 2016

Modifica	Descrizione	Data
Analisi di Amazon S3 – Analisi della classe di storage	La nuova caratteristica di analisi di Amazon S3 (analisi della classe di archiviazione) osserva i modelli di accesso ai dati per aiutare a determinare quando è opportuno spostare i dati meno utilizzati dalla classe di archiviazione `STANDARD` alla classe di archiviazione `STANDARD_IA` (IA sta per "Infrequent Access", accesso infrequente). Dopo l'osservazione degli schemi di accesso poco frequenti a un set di dati filtrati in un certo periodo di tempo da parte dell'analisi della classe di archiviazione, i risultati dell'anal isi possono essere utilizzati per migliorare le configura zioni del ciclo di vita. Questa caratteristica include anche un'analisi giornaliera dettagliata dell'utilizzo dello storage a livello di un bucket, prefisso o tag specificato che può essere esportata in un bucket S3.	29 Novembre 2016
Nuovo recupero rapido e in blocco dei dati durante il ripristino di oggetti archiviati da S3 Glacier	Amazon S3 supporta ora il recupero rapido e in blocco dei dati oltre al recupero standard durante il ripristino di oggetti archiviati in S3 Glacier. Per ulteriori informazioni, consulta Ripristino di un oggetto archiviato.	21 novembre 2016
CloudTrail registraz ione di oggetti	CloudTrail supporta la registrazione di operazioni API a livello di oggetto di Amazon S3 `GetObject` come`PutObject` , e. `DeleteObject` È possibile configurare alcuni selettori di evento per registrare le operazioni delle API a livello di oggetto. Per ulteriori informazioni, consulta Registrazione delle chiamate API Amazon S3 tramite AWS CloudTrail.	21 Novembre 2016
US East (Ohio) Region	Amazon S3 è ora disponibile nella regione Stati Uniti orientali (Ohio). Per ulteriori informazioni sulle regioni e gli endpoint Amazon S3, consultare la sezione relativa a regioni ed endpoint nella Riferimenti generali di AWS.	17 Ottobre 2016

Modifica	Descrizione	Data
Supporto di IPv6 per Amazon S3 Transfer Acceleration	Amazon S3 supporta ora IPv6 (Internet Protocol versione 6) per Amazon S3 Transfer Acceleration. È possibile connettersi ad Amazon S3 tramite IPv6 utilizzando il nuovo endpoint dual-stack per Transfer Acceleration. Per ulteriori informazioni, consulta Nozioni di base su Amazon S3 Transfer Acceleration.	6 ottobre 2016
Supporto IPv6	Amazon S3 supporta ora IPv6 (Internet Protocol versione 6). È possibile accedere ad Amazon S3 tramite IPv6 mediante gli endpoint dual-stack. Per ulteriori informazioni, consulta Esecuzione di richieste ad Amazon S3 su IPv6.	11 agosto 2016
Asia Pacific (Mumbai) Region	Amazon S3 è ora disponibile nella regione Asia Pacifico (Mumbai). Per ulteriori informazioni sulle regioni e gli endpoint Amazon S3, consultare la sezione relativa a regioni ed endpoint nella Riferimenti generali di AWS.	27 giugno 2016
Amazon S3 Transfer Acceleration	Amazon S3 Transfer Acceleration permette il trasferimento rapido, semplice e sicuro di file su lunga distanza tra un client e un bucket S3. Transfer Acceleration sfrutta le edge location distribuite su CloudFront scala globale di Amazon. Per ulteriori informazioni, consulta Configurazione di trasferimenti veloci e sicuri di file con Amazon S3 Transfer Acceleration.	19 Aprile 2016
Supporto del ciclo di vita per rimuovere i contrassegni di eliminazione degli oggetti scaduti	L'azione `Expiration` di configurazione del ciclo di vita consente ora di configurare Amazon S3 per rimuovere i contrassegni di eliminazione degli oggetti scaduti in un bucket con versione. Per ulteriori informazioni, consulta Elementi per la descrizione delle operazioni nel ciclo di vita.	16 marzo 2016

Modifica	Descrizione	Data
La configurazione del ciclo di vita del bucket supporta l'operazione per interrompere i caricamenti in più parti incompleti	La configurazione del ciclo di vita del bucket supporta ora l'operazione `AbortIncompleteMultipartUpload`, che è possibile utilizzare per fare in modo che Amazon S3 interrompa i caricamenti in più parti che non vengono completati entro un numero specificato di giorni dopo l'avvio. Quando un caricamento in più parti incompleto diventa idoneo per un'operazione di interruzione, Amazon S3 elimina le parti caricate e interrompe il caricamento in più parti. Per informazioni più tecniche, consulta gli argomenti seguenti nella Guida per l'utente di Amazon S3: • Interruzione di un caricamento in più parti • Elementi per la descrizione delle operazioni nel ciclo di vita Le seguenti operazioni API sono state aggiornata per supportare la nuova operazione: • PUT Bucket lifecycle (Ciclo di vita PUT Bucket) – La configurazione XML permette ora di specificare l'operazione `AbortIncompleteMultipartUpload` in una regola di configurazione del ciclo di vita. • List Parts (Elenco parti) e Initiate Multipart Upload (Avvio del caricamento in più parti) – Entrambe queste operazioni API restituiscono ora due intestazioni di risposta aggiuntive (`x-amz-abort-date` e `x-amz-abort-rule-id`) se al bucket è associata una regola del ciclo di vita che specifica l'operazione `AbortIncompleteMultipartUpload`. Queste intestazioni di risposta indicano in quale momento il caricamento in più	16 marzo 2016

Modifica	Descrizione	Data
	parti avviato è diventato idoneo all'operazione di interruzione e quale regola del ciclo di vita può essere applicata.	
Asia Pacific (Seoul) Region	Amazon S3 è ora disponibile nella regione Asia Pacifico (Seoul). Per ulteriori informazioni sulle regioni e gli endpoint Amazon S3, consultare la sezione relativa a regioni ed endpoint nella Riferimenti generali di AWS.	6 gennaio 2016
Nuova chiave di condizione e modifica del caricamento in più parti	Le policy IAM supportano ora una chiave di condizione `s3:x-amz-storage-class` Amazon S3. Per ulteriori informazioni, consulta Esempi di policy Bucket che utilizzano chiavi condizionali. Non è più necessario essere l'account che ha iniziato un caricamento in più parti per caricare le parti e completare il caricamento. Per ulteriori informazioni, consulta Autorizzazioni e API per il caricamento in più parti.	14 dicembre 2015
Regione Stati Uniti standard rinominata	È stata modificata la stringa del nome della regione da "Stati Uniti standard" a "Stati Uniti orientali (Virginia settentrionale)". Si tratta solo di un aggiornamento del nome della regione, senza modifiche della funzionalità.	11 dicembre 2015

Modifica	Descrizione	Data
Nuova classe di storage	Amazon S3 offre ora una nuova classe di storage, STANDARD_IA (IA sta per "Infrequent Access", accesso infrequente) per archiviare gli oggetti. Questa classe di storage è ottimizzata per i dati esistenti da molto tempo a cui si accede meno frequentemente. Per ulteriori informazioni, consulta Utilizzo delle classi di storage di Amazon S3. La funzione di configurazione del ciclo di vita consente ora la transizione degli oggetti alla classe di storage standard_IA. Per ulteriori informazioni, consulta Gestione del ciclo di vita dello storage. In precedenza, la funzione di replica tra regioni si serviva della classe di storage dell'oggetto di origine per la replica di oggetti. Ora, quando si configura la replica tra regioni, è possibile specificare una classe di storage per la replica dell'oggetto creata nel bucket di destinazione. Per ulteriori informazioni, consulta Panoramica sulla replica degli oggetti.	16 settembre 2015
AWS CloudTrail integrazione	La nuova AWS CloudTrail integrazione ti consente di registrare l'attività dell'API Amazon S3 nel tuo bucket S3. Puoi utilizzarlo CloudTrail per tenere traccia delle creazioni o delle eliminazioni dei bucket S3, delle modifiche al controllo degli accessi o delle modifiche alla configurazione del ciclo di vita. Per ulteriori informazioni, consulta Registrazione delle chiamate API Amazon S3 tramite AWS CloudTrail.	1 settembre 2015

Modifica	Descrizione	Data
Aumento del limite del bucket	Amazon S3 supporta ora l'aumento dei limiti per i bucket. Per impostazione predefinita, i clienti possono creare fino a 100 bucket al loro interno. Account AWS I clienti che hanno bisogno di altri bucket possono aumentare tale limite richiedendo un aumento del limite di servizio. Per informazioni su come aumentare il limite per i bucket, consulta Service Quotas di Servizio AWS nella Documentazione di riferimento generale di AWS . Per ulteriori informazioni, consulta AWS Utilizzo degli SDK e Restrizioni e limitazioni dei bucket.	4 agosto 2015
Aggiornamento del modello di consistenza	Amazon S3 ora supporta la read-after-write coerenza per i nuovi oggetti aggiunti ad Amazon S3 nella regione Stati Uniti orientali (Virginia settentrionale). Prima di questo aggiornamento, tutte le regioni tranne la regione Stati Uniti orientali (Virginia settentrionale) supportavano la read-after-write coerenza per i nuovi oggetti caricati su Amazon S3. Con questo miglioramento, Amazon S3 ora read-after-write supporta la coerenza in tutte le regioni per i nuovi oggetti aggiunti ad Amazon S3. R ead-after-write consistency consente di recuperare gli oggetti immediatamente dopo la creazione in Amazon S3. Per ulteriori informazioni, consulta Regioni.	4 agosto 2015
Notifiche eventi	La funzionalità di notifica eventi di Amazon S3 è stata aggiornata con l'aggiunta di notifiche quando gli oggetti vengono eliminati e l'aggiunta di filtri basati sui nomi degli oggetti con prefisso e suffisso corrispondenti. Per ulteriori informazioni, consultare Notifiche di eventi Amazon S3.	28 luglio 2015

Modifica	Descrizione	Data
CloudWatch Integrazione con Amazon	La nuova CloudWatch integrazione con Amazon ti consente di monitorare e impostare allarmi sull'utilizzo di Amazon S3 CloudWatch tramite parametri per Amazon S3. I parametri supportati includono i byte totali per l'archiviazione standard, i byte totali per Reduced Redundancy Storage (RRS) e il numero totale di oggetti per un bucket S3 specifico. Per ulteriori informazioni, consultare Monitoraggio delle metriche con Amazon CloudWatch.	28 luglio 2015
Supporto per eliminare e svuotare i bucket non vuoti	Amazon S3 supporta ora l'eliminazione e lo svuotamento dei bucket non vuoti. Per ulteriori informazioni, consulta Svuotamento di un bucket.	16 luglio 2015
Policy di bucket per gli endpoint di Amazon VPC	Amazon S3 ha aggiunto il supporto per le policy del bucket per gli endpoint Virtual Private Cloud (VPC). È possibile utilizzare le policy del bucket S3 per controllare l'accesso ai bucket da VPC o da endpoint VPC specifici. Gli endpoint VPC sono facili da configurare ed estremamente affidabili e offrono una connessione sicura ad Amazon S3 senza la necessità di un gateway o di un'istanza NAT. Per ulteriori informazioni, consulta Controllo dell'accesso dagli endpoint VPC con policy di bucket.	29 Aprile 2015
Notifiche degli eventi	Le notifiche degli eventi di Amazon S3 sono state aggiornate per supportare il passaggio alle autorizzazioni basate sulle risorse per le funzioni. AWS Lambda Per ulteriori informazioni, consulta Notifiche di eventi Amazon S3.	9 Aprile 2015
Replica tra regioni	Amazon S3 supporta ora la replica tra regioni. La replica interregionale è la copia automatica e asincrona di oggetti tra bucket diversi. Regioni AWS Per ulteriori informazioni, consulta Panoramica sulla replica degli oggetti.	24 marzo 2015

Modifica	Descrizione	Data
Notifiche eventi	Amazon S3 supporta ora nuovi tipi di evento e destinazioni nella configurazione delle notifiche dei bucket. Prima di questa versione, Amazon S3 supportava solo il tipo di ReducedRedundancyLostObject evento s3: e un argomento Amazon SNS come destinazione. Per ulteriori informazioni sull'utilizzo dei nuovi tipi di eventi, consulta Notifiche di eventi Amazon S3.	13 Novembre 2014
Crittografia lato server con chiavi di crittografia fornire dal cliente	Crittografia lato server con chiavi () (AWS Key Management Service SSE-KMS)AWS KMS Amazon S3 ora supporta la crittografia lato server utilizzando. AWS KMS Questa funzionalità consente di gestire la chiave della busta tramite AWS KMS e le AWS KMS chiamate Amazon S3 per accedere alla chiave della busta entro le autorizzazioni impostate. Per ulteriori informazioni sulla crittografia lato server con AWS KMS, consulta Protezione dei dati utilizzando la crittografia lato server con. AWS Key Management Service	12 Novembre 2014
Europe (Frankfurt) Region	Amazon S3 è ora disponibile nella regione UE (Francoforte).	23 ottobre 2014

Modifica	Descrizione	Data
Crittografia lato server con chiavi di crittografia fornire dal cliente	Amazon S3 supporta ora la crittografia lato server con chiavi di crittografia fornite dal cliente (SSE-C). La crittografia lato server permette di richiedere ad Amazon S3 di crittografare i dati inattivi. Quando si utilizza SSE-C, Amazon S3 esegue la crittografia degli oggetti con le chiavi di crittografia personalizzate fornite. Poiché Amazon S3 esegue automaticamente la crittografia, si ottengono i vantaggi legati all'utilizzo di chiavi di crittografia personali, ma senza che sia necessario scrivere o eseguire codice di crittografia personalizzato. Per ulteriori informazioni su SSE-C, consulta Protezione dei dati con la crittografia lato server con chiavi di crittografia fornite dal cliente (SSE-C).	12 giugno 2014
Supporto della funzione Controllo delle versioni a livello del ciclo di vita	Nelle release precedenti, la configurazione del ciclo di vita era supportata solo per i bucket senza versione. Ora è possibile configurare il ciclo di vita sia sui bucket senza versione che su quelli con funzione Controllo delle versioni abilitata. Per ulteriori informazioni, consulta Gestione del ciclo di vita dello storage.	20 maggio 2014
Argomenti del controllo degli accessi modificati	È stata modificata la documentazione sul controllo degli accessi di Amazon S3. Per ulteriori informazioni, consulta Identity and Access Management per Amazon S3.	15 Aprile 2014
Argomenti relativi alla registrazione degli accessi al server modificati	È stata modificata la documentazione sulla registrazione degli accessi al server. Per ulteriori informazioni, consulta Registrazione delle richieste con registrazione dell'accesso al server.	26 Novembre 2013
Esempi sull'SDK .NET aggiornati alla versione 2.0	Gli esempi sull'SDK .NET di questa guida sono ora conformi alla versione 2.0.	26 Novembre 2013

Modifica	Descrizione	Data
Supporto SOAP su HTTP obsoleto	Il supporto di SOAP su HTTP non viene più utilizzato, ma è ancora disponibile su HTTPS. Le nuove funzioni di Amazon S3 non saranno supportate per SOAP. Ti consigliamo di utilizzare l'API REST o gli SDK. AWS	20 settembre 2013
Supporto delle variabili di policy IAM	Il linguaggio delle policy IAM ora supporta le variabili . Quando una policy viene valutata, eventuali variabili di policy vengono sostituite con valori forniti in base a informazioni basate sul contesto relative alla sessione dell'utente autenticato. Si possono utilizzare variabili di policy per definire policy generali senza elencare esplicitamente tutte le componenti della policy. Per ulteriori informazioni sulle variabili di policy, consulta la pagina relativa alla panoramica delle variabili di policy IAM nella Guida per l'utente di IAM. Per esempi di variabili di policy in Amazon S3, consulta Esempi di policy basate sull'identità per Amazon S3.	3 Aprile 2013
Supporto di Pagamento a carico del richiedente a livello console	È ora possibile configurare il bucket per Pagamento a carico del richiedente tramite la console Amazon S3. Per ulteriori informazioni, consulta Utilizzo dei bucket con pagamento a carico del richiedente per utilizzo e trasferimenti di storage.	31 dicembre 2012

Modifica	Descrizione	Data
Supporto per l'hosting di un sito Web in un dominio root	Amazon S3 supporta ora l'hosting di siti Web statici in un dominio root. I visitatori del sito Web possono accedere al sito dal loro browser senza specificare www nell'indirizzo Web (ad esempio, digitando example.com al posto di www.example.com). Molti clienti ospitano già siti Web statici in Amazon S3 accessibili tramite un sottodominio www, ad esempio www.example.com. In precedenza, per supportare l'accesso al dominio root era necessario eseguire il proprio server Web per inoltrare tramite proxy le richieste del dominio root dai browser al sito Web in Amazon S3. L'utilizzo di un server Web per le richieste proxy comporta ulteriori costi, carichi operativi e un'altra possibilità di errore. Ora è possibile sfruttare i vantaggi della durabilità e della disponibilità elevata di Amazon S3 sia per gli indirizzi di dominio root che www. Per ulteriori informazioni, consulta Hosting di un sito Web statico tramite Amazon S3.	27 dicembre 2012
Revisione della console	La console Amazon S3 è stata aggiornata. Gli argomenti della documentazione che si riferiscono alla console sono stati modificati di conseguenza.	14 dicembre 2012
Supporto per l'archiviazione dei dati in S3 Glacier	Amazon S3 supporta ora un'opzione di storage che permette di utilizzare il servizio di storage a basso costo di S3 Glacier per l'archiviazione dei dati. Per archiviare gli oggetti, è necessario definire le regole di archiviazione che identificano gli oggetti e l'intervallo di tempo in cui Amazon S3 deve archiviare questi oggetti in S3 Glacier. Puoi impostare facilmente le regole su un bucket utilizzando la console Amazon S3 o a livello di codice utilizzando l'API o gli SDK di Amazon S3. AWS Per ulteriori informazioni, consulta Gestione del ciclo di vita dello storage.	13 Novembre 2012

Modifica	Descrizione	Data
Supporto del reindirizzamento delle pagine del sito Web	Per i bucket configurati come sito Web, Amazon S3 supporta ora il reindirizzamento di una richiesta di un oggetto a un altro oggetto nello stesso bucket o a un URL esterno. Per ulteriori informazioni, consulta (Facoltativo) Configurazione del reindirizzamento di una pagina Web. Per informazioni sull'hosting dei siti Web, consulta Hosting di un sito Web statico tramite Amazon S3.	4 Ottobre 2012
Supporto di Cross Origin Resource Sharing (CORS)	Amazon S3 supporta ora CORS (Cross-Origin Resource Sharing, condivisione delle risorse multiorigine). La funzionalità CORS definisce un metodo con cui le applicazioni Web dei clienti caricate in un dominio possono interagire con le risorse situate in un dominio differente o accedere a esse. Con il supporto di CORS in Amazon S3 è possibile creare applicazioni Web lato client complete basate su Amazon S3 e concedere l'accesso selettivo tra domini alle risorse di Amazon S3. Per ulteriori informazioni, consulta Utilizzo della funzionalità Cross-Origin Resource Sharing (CORS).	31 agosto 2012
Supporto dei tag dell'allocazione dei costi	Amazon S3 supporta ora l'assegnazione di tag per l'allocazione dei costi, che permette di etichettare i bucket S3 in modo da poter tenere traccia più facilmente dei costi in base ai progetti o ad altri criteri. Per ulteriori informazioni sull'utilizzo del tagging per i bucket, consulta Utilizzo dei tag per l'allocazione dei costi per i bucket S3.	21 agosto 2012

Modifica	Descrizione	Data
Supporto per l'accesso all'API protetto con autentica zione MFA nelle policy di bucket	Amazon S3 ora supporta l'accesso alle API protetto da MFA, una funzionalità che può applicare l' AWS autentica zione a più fattori per un ulteriore livello di sicurezza durante l'accesso alle risorse Amazon S3. È una funzione di protezione che prevede che gli utenti dimostrino di possedere fisicamente un dispositivo MFA fornendo un codice MFA valido. Per ulteriori informazioni, consulta Autenticazione a più fattori (MFA) di AWS. È ora possibile richiedere l'autenticazione MFA per tutte le richieste di accesso alle risorse di Amazon S3. Per imporre il requisito dell'autenticazione MFA, Amazon S3 supporta ora la chiave `aws:MultiFactorAuthAge` in una policy del bucket. Per un esempio di policy di bucket, consulta Richiesta dell'autenticazione a più fattori (MFA).	10 luglio 2012
Supporto per la scadenza degli oggetti	Si può utilizzare la scadenza degli oggetti per pianificare la rimozione automatica dei dati dopo un periodo di tempo configurato. Per impostare la scadenza degli oggetti, devi aggiungere la configurazione del ciclo di vita in un bucket.	27 dicembre 2011
Nuova regione supportata	Amazon S3 supporta ora la regione Sud America (San Paolo). Per ulteriori informazioni, consulta Accesso ed elenco di un bucket Amazon S3.	14 dicembre 2011
Eliminazione di più oggetti	Amazon S3 supporta ora l'API Multi-Object Delete che permette di eliminare più oggetti in una singola richiesta. Grazie a questa caratteristica, è possibile rimuovere grandi quantità di oggetti da Amazon S3 più rapidamente rispetto all'utilizzo di più richieste DELETE singole. Per ulteriori informazioni, consulta Eliminazione di oggetti Amazon S3 .	7 dicembre 2011
Nuova regione supportata	Amazon S3 supporta ora la regione Stati Uniti occidenta li (Oregon). Per ulteriori informazioni, consulta Bucket e regioni.	8 Novembre 2011

Modifica	Descrizione	Data
Aggiornamento della documentazione	Correzione dei bug della documentazione.	8 Novembre 2011
Aggiornamento della documentazione	Oltre alla correzione dei bug della documentazione, questa release include i seguenti miglioramenti: • Nuove sezioni di crittografia lato server che utilizzano e (vedi). AWS SDK for PHP AWS SDK for Ruby Specifica della crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3)	17 ottobre 2011
Supporto per la crittografia lato server	Amazon S3 supporta ora la crittografia lato server. Permette di richiedere ad Amazon S3 di crittografare i dati inattivi, ovvero i dati degli oggetti quando Amazon S3 scrive i dati sui dischi nei data center. Oltre agli aggiornam enti delle API REST, e.NET forniscono AWS SDK for Java le funzionalità necessarie per richiedere la crittogra fia lato server. È anche possibile richiedere la crittogra fia lato server durante il caricamento di oggetti tramite la AWS Management Console. Per ulteriori informazioni sulla crittografia dei dati, consulta Utilizzo della crittografia dei dati.	4 ottobre 2011

Modifica	Descrizione	Data
Aggiornamento della documentazione	Oltre alla correzione dei bug della documentazione, questa release include i seguenti miglioramenti: • Sono stati aggiunti esempi su Ruby e PHP nella sezione Esecuzione di richieste. • Aggiunta di sezioni che descrivono come generare e usare URL prefirmati. Per ulteriori informazioni, consulta Condivisione di oggetti mediante URL prefirmati e Condivisione di oggetti mediante URL prefirmati. • È stata aggiornata una sezione esistente per introdurre AWS Explorers for Eclipse e Visual Studio. Per ulteriori informazioni, consulta Sviluppo con Amazon S3 utilizzando gli SDK AWS.	22 settembre 2011

Modifica	Descrizione	Data
Supporto all'invio di richieste mediante credenziali di sicurezza temporanee	Oltre a utilizzare le tue credenziali di sicurezza utente Account AWS e IAM per inviare richieste autenticate ad Amazon S3, ora puoi inviare richieste utilizzando credenziali di sicurezza temporanee AWS Identity and Access Management ottenute da (IAM). Puoi utilizzare l' AWS Security Token Service API o le librerie wrapper AWS SDK per richiedere queste credenziali temporanee a IAM. Si possono richiedere queste credenziali di sicurezza temporanee per uso personale oppure per fornirle agli utenti federati e alle applicazioni. Questa funzionalità ti consente di gestire gli utenti all'esterno AWS e di fornire loro credenziali di sicurezza temporanee per accedere alle tue risorse. AWS Per ulteriori informazioni, consulta Esecuzione di richieste. Per ulteriori informazioni sul supporto IAM per le credenziali di sicurezza temporanee, consulta la sezione relativa alle credenziali di sicurezza temporanee nella Guida per l'utente di IAM.	3 agosto 2011
L'API per il caricamento in più parti è stata ampliata per consentire la copia di oggetti fino a 5 TB	Prima di questa versione, l'API Amazon S3 supportava la copia di oggetti con dimensioni massime di 5 GB. Per permettere la copia di oggetti con dimensioni superiori a 5 GB, Amazon S3 ora estende l'API per il caricamento in più parti con una nuova operazione, `Upload Part (Copy)` . È possibile utilizzare questa operazione di caricamento in più parti per copiare gli oggetti con dimensioni massime di 5 TB. Per ulteriori informazioni, consulta Copiare, spostare e rinominare oggetti. Per informazioni più tecniche sull'API per il caricamento in più parti, consulta Caricamento e copia di oggetti utilizzando il caricamento in più parti.	21 giugno 2011

Modifica	Descrizione	Data
Chiamate all'API SOAP su HTTP disabilitate	Per aumentare la sicurezza, le chiamate all'API SOAP su HTTP sono state disabilitate. Le richieste SOAP autenticate e anonime devono essere inviate ad Amazon S3 tramite SSL.	6 giugno 2011
IAM permette la delega multiaccount	In precedenza, per accedere a una risorsa Amazon S3, un utente IAM aveva bisogno delle autorizzazioni sia del genitore che del proprietario Account AWS della risorsa Amazon S3. Con l'accesso multiaccount l'utente IAM deve ora disporre solo dell'autorizzazione concessa dall'account proprietario. Cioè, se il proprietario di una risorsa concede l'accesso a una Account AWS, ora Account AWS può concedere ai suoi utenti IAM l'accesso a tali risorse. Per ulteriori informazioni, consulta la pagina relativa alla creazione di un ruolo per delegare le autorizzazioni a un utente IAM nella Guida per l'utente IAM. Per ulteriori informazioni sulla specifica delle informazioni principali nella policy di un bucket, consulta Principi per le policy relative ai bucket.	6 giugno 2011
Nuovo collegamento	Le informazioni relative all'endpoint del servizio si trovano ora nella Documentazione di riferimento generale di AWS. Per ulteriori informazioni, consulta la pagina relativa a regioni ed endpoint nella Documentazione di riferimento generale di AWS.	1 marzo 2011

Modifica	Descrizione	Data
Supporto per l'hosting di siti Web statici in Amazon S3	Amazon S3 introduce il supporto migliorato per l'hosting di siti Web statici. È incluso il supporto per i documenti di indice e i documenti di errore personalizzati. Quando utilizzi queste funzionalità, le richieste alla root del bucket o a una cartella (ad esempio `http://mywebsite.com/subfolder`) restituiscono il documento di indice invece dell'elenco di oggetti nel bucket. Se si verifica un errore, Amazon S3 restituisce il messaggio di errore personalizzato anziché un messaggio di errore di Amazon S3. Per ulteriori informazioni, consulta Hosting di un sito Web statico tramite Amazon S3.	6 giugno 2011
Le informazioni relative all'endpoint del servizio si trovano ora nella Documentazione di riferimento generale di AWS . Per ulteriori informazioni, consulta la pagina relativa a regioni ed endpoint nella Documentazione di riferimento generale di AWS.	1 marzo 2011	

Modifica	Descrizione	Data
Supporto per l'hosting di siti Web statici in Amazon S3	Amazon S3 introduce il supporto migliorato per l'hosting di siti Web statici. È incluso il supporto per i documenti di indice e i documenti di errore personalizzati. Quando utilizzi queste funzionalità, le richieste alla root del bucket o a una cartella (ad esempio `http://mywebsite.com/subfolder`) restituiscono il documento di indice invece dell'elenco di oggetti nel bucket. Se si verifica un errore, Amazon S3 restituisce il messaggio di errore personalizzato anziché un messaggio di errore di Amazon S3. Per ulteriori informazioni, consulta Hosting di un sito Web statico tramite Amazon S3.	17 febbraio 2011
Supporto API per le intestazioni di risposta	L'API GET Object REST ora consente di modificare le intestazioni di risposta di ogni richiesta REST GET Object. Si possono quindi modificare i metadata degli oggetti nella risposta senza modificare l'oggetto in sé. Per ulteriori informazioni, consulta Download di oggetti.	14 gennaio 2011
Supporto ampliato per gli oggetti	Amazon S3 ha aumentato le dimensioni massime di un oggetto che è possibile archiviare in un bucket S3 da 5 GB a 5 TB. Se si utilizza la REST API, è possibile caricare oggetti con una dimensione massima di 5 GB con una singola operazione PUT. Per oggetti di dimensioni maggiori occorre utilizzare l'API REST per il caricamento in più parti. Per ulteriori informazioni, consulta Caricamento e copia di oggetti utilizzando il caricamento in più parti.	9 dicembre 2010
Caricamento in più parti	Il caricamento in più parti permette operazioni di caricamento più rapide e flessibili in Amazon S3. Permette di caricare un unico oggetto come un insieme di parti. Per ulteriori informazioni, consulta Caricamento e copia di oggetti utilizzando il caricamento in più parti.	10 Novembre 2010

Modifica	Descrizione	Data
Supporto ID canonico nelle policy di bucket	È ora possibile specificare ID convenzionali nelle policy di bucket. Per ulteriori informazioni, consulta Principi per le policy relative ai bucket	17 settembre 2010
Amazon S3 funziona con IAM	Questo servizio ora si integra con AWS Identity and Access Management (IAM). Per ulteriori informazioni, consulta Servizi AWS supportati da IAM nella Guida per l'utente di IAM.	2 settembre 2010
Notifiche	La caratteristica di notifica Amazon S3 permette di configurare un bucket in modo che Amazon S3 pubblichi un messaggio in un argomento Amazon Simple Notification Service (Amazon SNS) quando Amazon S3 rileva un evento chiave in un bucket. Per ulteriori informazioni, consulta Configurazione delle notifiche degli eventi del bucket.	14 luglio 2010
Policy di bucket	Le policy di bucket sono un sistema di gestione degli accessi utilizzato per configurare le autorizzazioni di accesso a bucket, oggetti e insiemi di oggetti. Questa funzionalità integra, e in molti casi sostituisce, le liste di controllo accessi. Per ulteriori informazioni, consulta Politiche Bucket per Amazon S3.	6 luglio 2010
Sintassi basata su percorsi disponibile in tutte le regioni	Amazon S3 supporta ora la sintassi basata su percorsi per qualsiasi bucket nella regione Stati Uniti classica oppure se il bucket si trova nella stessa regione dell'endpoint della richiesta. Per ulteriori informazioni, consulta Hosting virtuale.	9 giugno 2010
Nuovo endpoint per la regione UE (Irlanda)	Amazon S3 fornisce ora un endpoint per la regione UE (Irlanda): `http://s3-eu-west-1.amazonaws.com` .	9 giugno 2010

Modifica	Descrizione	Data
Console	È ora possibile utilizzare Amazon S3 tramite la AWS Management Console. Puoi trovare le informazioni su tutte le funzionalità di Amazon S3 nella console nella Guida per l'utente di Amazon Simple Storage Service.	9 giugno 2010
Ridondanza ridotta	Amazon S3 permette ora di ridurre i costi di storage tramite l'archiviazione degli oggetti in Amazon S3 con ridondanz a ridotta. Per ulteriori informazioni, consulta Reduced Redundancy Storage.	12 maggio 2010
Nuova regione supportata	Amazon S3 supporta ora la regione Asia Pacifico (Singapor e). Per ulteriori informazioni, consulta Bucket e regioni.	28 Aprile 2010
Funzione Controllo delle versioni degli oggetti	In questa release è stata introdotta la funzione Controllo delle versioni degli oggetti. Tutti gli oggetti possono ora avere una chiave e una versione. Se si abilita la funzione Controllo delle versioni del bucket, Amazon S3 fornisce a tutti gli oggetti aggiunti a tale bucket un ID versione univoco. Questa caratteristica ti permette di eseguire il ripristino in seguito a operazioni accidentali di eliminazi one e sovrascrittura. Per ulteriori informazioni, consulta Funzione Controllo delle versioni e Uso della funzione Controllo delle versioni.	8 febbraio 2010
Nuova regione supportata	Amazon S3 supporta ora la regione Stati Uniti occidentali (California settentrionale). Il nuovo endpoint per le richieste a questa regione è `s3-us-west-1.amazonaws.com` . Per ulteriori informazioni, consulta Bucket e regioni.	2 dicembre 2009

Modifica	Descrizione	Data
AWS SDK for .NET	AWS ora fornisce librerie, codice di esempio, tutorial e altre risorse per gli sviluppatori di software che preferiscono creare applicazioni utilizzando operazioni API specifiche del linguaggio.NET anziché REST o SOAP. Queste librerie forniscono funzioni di base (non incluse nelle API REST o SOAP), come autenticazione di richiesta, nuovi tentativi di richiesta e gestione degli errori per iniziare in modo più semplice. Per ulteriori informazioni sulle librerie e risorse specifiche di un linguaggio, consulta Sviluppo con Amazon S3 utilizzando gli SDK AWS.	11 Novembre 2009

Glossario per AWS

Per la terminologia AWS più recente, consultare il glossario AWS nella documentazione di riferimento per Glossario AWS.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.