Come StorageGRID implementa l'API REST S3

StorageGRID software

NetApp
December 03, 2025

Sommario

Come StorageGRID implementa l'API REST S3

Richieste dei clienti in conflitto

Le richieste dei client in conflitto, ad esempio due client che scrivono sulla stessa chiave, vengono risolte in base al principio "latest-wins".

La tempistica per la valutazione "latest-wins" si basa sul momento in cui il sistema StorageGRID completa una determinata richiesta e non su quando i client S3 iniziano un'operazione.

Valori di coerenza

La coerenza fornisce un equilibrio tra la disponibilità degli oggetti e la coerenza di tali oggetti tra diversi nodi di archiviazione e siti. È possibile modificare la coerenza in base alle esigenze dell'applicazione.

Per impostazione predefinita, StorageGRID garantisce la coerenza di lettura dopo scrittura per gli oggetti appena creati. Qualsiasi GET successivo a un PUT completato con successo sarà in grado di leggere i dati appena scritti. Le sovrascritture di oggetti esistenti, gli aggiornamenti dei metadati e le eliminazioni alla fine risultano coerenti. In genere, la propagazione delle sovrascritture richiede secondi o minuti, ma può richiedere fino a 15 giorni.

Se si desidera eseguire operazioni sugli oggetti con una consistenza diversa, è possibile:

- Specificare una coerenza perogni secchio .
- Specificare una coerenza perogni operazione API .
- Modificare la coerenza predefinita dell'intera griglia eseguendo una delle seguenti attività:
  - In Grid Manager, vai su **CONFIGURAZIONE** > **Sistema** > **Impostazioni di archiviazione** > **Coerenza predefinita**.
  - .

> Una modifica alla coerenza a livello di griglia si applica solo ai bucket creati dopo la modifica dell'impostazione. Per determinare i dettagli di una modifica, consultare il registro di controllo situato in `/var/local/log` (cerca **consistencyLevel**).

Valori di coerenza

La coerenza influisce sul modo in cui i metadati utilizzati da StorageGRID per tracciare gli oggetti vengono distribuiti tra i nodi e, di conseguenza, sulla disponibilità degli oggetti per le richieste dei client.

È possibile impostare la coerenza per un bucket o un'operazione API su uno dei seguenti valori:

- **Tutti**: tutti i nodi ricevono immediatamente i dati, altrimenti la richiesta fallirà.
- **Strong-global**: garantisce la coerenza di lettura e scrittura per tutte le richieste dei client su tutti i siti.
- **Strong-site**: garantisce la coerenza di lettura e scrittura per tutte le richieste client all'interno di un sito.
- **Lettura dopo nuova scrittura**: (predefinito) fornisce coerenza di lettura dopo scrittura per i nuovi oggetti e coerenza finale per gli aggiornamenti degli oggetti. Offre elevate garanzie di disponibilità e protezione dei

dati. Consigliato nella maggior parte dei casi.

- **Disponibile**: fornisce coerenza finale sia per i nuovi oggetti che per gli aggiornamenti degli oggetti. Per i bucket S3, utilizzare solo se necessario (ad esempio, per un bucket che contiene valori di log che vengono letti raramente o per operazioni HEAD o GET su chiavi inesistenti). Non supportato per i bucket S3 FabricPool .

Utilizzare la coerenza "Lettura dopo nuova scrittura" e "Disponibile"

Quando un'operazione HEAD o GET utilizza la coerenza "Read-after-new-write", StorageGRID esegue la ricerca in più passaggi, come segue:

- Per prima cosa cerca l'oggetto utilizzando una bassa coerenza.
- Se la ricerca fallisce, la ripete al valore di coerenza successivo finché non raggiunge una coerenza equivalente al comportamento di strong-global.

Se un'operazione HEAD o GET utilizza la coerenza "Read-after-new-write" ma l'oggetto non esiste, la ricerca dell'oggetto raggiungerà sempre una coerenza equivalente al comportamento per strong-global. Poiché questa coerenza richiede che siano disponibili più copie dei metadati dell'oggetto in ogni sito, è possibile ricevere un numero elevato di errori 500 Internal Server se due o più nodi di archiviazione nello stesso sito non sono disponibili.

A meno che non siano necessarie garanzie di coerenza simili ad Amazon S3, è possibile prevenire questi errori per le operazioni HEAD e GET impostando la coerenza su "Disponibile". Quando un'operazione HEAD o GET utilizza la coerenza "Disponibile", StorageGRID fornisce solo la coerenza finale. Non riprova un'operazione non riuscita aumentando la coerenza, quindi non richiede che siano disponibili più copie dei metadati dell'oggetto.

Specifica la coerenza per l'operazione API

Per impostare la coerenza per una singola operazione API, i valori di coerenza devono essere supportati per l'operazione ed è necessario specificare la coerenza nell'intestazione della richiesta. In questo esempio la coerenza viene impostata su "Strong-site" per un'operazione GetObject.

```
GET /bucket/object HTTP/1.1
Date: date
Authorization: authorization name
Host: host
Consistency-Control: strong-site
```

È necessario utilizzare la stessa coerenza per entrambe le operazioni PutObject e GetObject.

Specifica la coerenza per il bucket

Per impostare la coerenza per il bucket, puoi utilizzare StorageGRID"PUT Consistenza del secchio" richiesta. Oppure puoi"modificare la consistenza di un bucket" dal gestore dell'affitto.

Quando si imposta la consistenza di un bucket, tenere presente quanto segue:

- L'impostazione della coerenza per un bucket determina quale coerenza viene utilizzata per le operazioni S3 eseguite sugli oggetti nel bucket o sulla configurazione del bucket. Non influisce sulle operazioni sul

bucket stesso.

- La coerenza di una singola operazione API prevale sulla coerenza del bucket.
- In generale, i bucket dovrebbero utilizzare la coerenza predefinita, "Lettura dopo nuova scrittura". Se le richieste non funzionano correttamente, modificare, se possibile, il comportamento del client dell'applicazione. Oppure, configura il client in modo che specifichi la coerenza per ogni richiesta API. Impostare la coerenza a livello di bucket solo come ultima risorsa.

Come interagiscono la coerenza e le regole ILM per influenzare la protezione dei dati

Sia la scelta della coerenza sia la regola ILM influiscono sul modo in cui gli oggetti vengono protetti. Queste impostazioni possono interagire.

Ad esempio, la coerenza utilizzata quando un oggetto viene archiviato influisce sul posizionamento iniziale dei metadati dell'oggetto, mentre il comportamento di acquisizione selezionato per la regola ILM influisce sul posizionamento iniziale delle copie dell'oggetto. Poiché StorageGRID richiede l'accesso sia ai metadati di un oggetto sia ai suoi dati per soddisfare le richieste dei client, la selezione di livelli di protezione corrispondenti per la coerenza e il comportamento di acquisizione può garantire una migliore protezione iniziale dei dati e risposte di sistema più prevedibili.

Il seguente"opzioni di ingestione" sono disponibili per le regole ILM:

Doppio impegno
: StorageGRID crea immediatamente copie provvisorie dell'oggetto e restituisce l'esito positivo al client. Quando possibile, vengono effettuate le copie specificate nella norma ILM.

Rigoroso
: Tutte le copie specificate nella regola ILM devono essere effettuate prima che il successo venga restituito al cliente.

Equilibrato
: StorageGRID tenta di effettuare tutte le copie specificate nella regola ILM al momento dell'acquisizione; se ciò non è possibile, vengono effettuate copie provvisorie e il client riceve un messaggio di conferma dell'operazione riuscita. Quando possibile, vengono effettuate le copie specificate nella norma ILM.

Esempio di come la coerenza e la regola ILM possono interagire

Supponiamo di avere una griglia a due siti con la seguente regola ILM e la seguente coerenza:

- **Regola ILM**: creare due copie dell'oggetto, una nel sito locale e una in un sito remoto. Utilizzare un comportamento di acquisizione rigoroso.
- **coerenza**: Strong-global (i metadati degli oggetti vengono distribuiti immediatamente a tutti i siti).

Quando un client memorizza un oggetto nella griglia, StorageGRID esegue entrambe le copie dell'oggetto e distribuisce i metadati a entrambi i siti prima di restituire l'esito positivo al client.

L'oggetto è completamente protetto contro la perdita al momento dell'acquisizione corretta del messaggio. Ad esempio, se il sito locale viene perso poco dopo l'acquisizione, copie sia dei dati dell'oggetto sia dei metadati dell'oggetto sono ancora presenti nel sito remoto. L'oggetto è completamente recuperabile.

Se invece si utilizzasse la stessa regola ILM e la coerenza del sito forte, il client potrebbe ricevere un messaggio di successo dopo che i dati dell'oggetto sono stati replicati sul sito remoto, ma prima che i metadati dell'oggetto vengano distribuiti lì. In questo caso, il livello di protezione dei metadati degli oggetti non

corrisponde al livello di protezione dei dati degli oggetti. Se il sito locale viene perso subito dopo l'acquisizione, anche i metadati dell'oggetto vengono persi. L'oggetto non può essere recuperato.

L'interrelazione tra coerenza e regole ILM può essere complessa. Se hai bisogno di assistenza, contatta NetApp .

Versionamento degli oggetti

È possibile impostare lo stato di controllo delle versioni di un bucket se si desidera conservare più versioni di ciascun oggetto. Abilitare il controllo delle versioni per un bucket può contribuire a proteggere gli oggetti dall'eliminazione accidentale e consente di recuperare e ripristinare le versioni precedenti di un oggetto.

Il sistema StorageGRID implementa il controllo delle versioni con supporto per la maggior parte delle funzionalità e con alcune limitazioni. StorageGRID supporta fino a 10.000 versioni di ciascun oggetto.

Il controllo delle versioni degli oggetti può essere combinato con la gestione del ciclo di vita delle informazioni (ILM) StorageGRID o con la configurazione del ciclo di vita del bucket S3. È necessario abilitare esplicitamente il controllo delle versioni per ogni bucket. Quando il controllo delle versioni è abilitato per un bucket, a ogni oggetto aggiunto al bucket viene assegnato un ID versione, generato dal sistema StorageGRID .

L'eliminazione tramite MFA (autenticazione a più fattori) non è supportata.

> Il controllo delle versioni può essere abilitato solo sui bucket creati con StorageGRID versione 10.3 o successiva.

ILM e controllo delle versioni

Le policy ILM vengono applicate a ciascuna versione di un oggetto. Un processo di scansione ILM analizza continuamente tutti gli oggetti e li rivaluta in base alla politica ILM corrente. Tutte le modifiche apportate ai criteri ILM vengono applicate a tutti gli oggetti precedentemente acquisiti. Sono incluse le versioni precedentemente acquisite se è abilitato il controllo delle versioni. La scansione ILM applica le nuove modifiche ILM agli oggetti precedentemente acquisiti.

Per gli oggetti S3 nei bucket abilitati per il controllo delle versioni, il supporto del controllo delle versioni consente di creare regole ILM che utilizzano "Ora non corrente" come ora di riferimento (selezionare **Sì** alla domanda "Applicare questa regola solo alle versioni precedenti degli oggetti?" in"Passaggio 1 della procedura guidata Crea una regola ILM"). Quando un oggetto viene aggiornato, le sue versioni precedenti diventano non aggiornate. Utilizzando un filtro "Tempo non corrente" è possibile creare criteri che riducono l'impatto sull'archiviazione delle versioni precedenti degli oggetti.

> Quando si carica una nuova versione di un oggetto utilizzando un'operazione di caricamento multiparte, il tempo non corrente per la versione originale dell'oggetto riflette il momento in cui è stato creato il caricamento multiparte per la nuova versione, non il momento in cui è stato completato il caricamento multiparte. In casi limitati, l'orario non aggiornato della versione originale potrebbe essere precedente di ore o giorni rispetto all'orario della versione corrente.

Informazioni correlate

- "Come vengono eliminati gli oggetti con versione S3"
- "Regole e policy ILM per oggetti con versione S3 (esempio 4)" .

Utilizzare l'API REST S3 per configurare S3 Object Lock

Se l'impostazione globale S3 Object Lock è abilitata per il sistema StorageGRID , è possibile creare bucket con S3 Object Lock abilitato. È possibile specificare la conservazione predefinita per ogni bucket o le impostazioni di conservazione per ogni versione dell'oggetto.

Come abilitare S3 Object Lock per un bucket

Se l'impostazione globale S3 Object Lock è abilitata per il sistema StorageGRID , è possibile abilitare facoltativamente S3 Object Lock quando si crea ogni bucket.

S3 Object Lock è un'impostazione permanente che può essere abilitata solo quando si crea un bucket. Non è possibile aggiungere o disabilitare S3 Object Lock dopo aver creato un bucket.

Per abilitare S3 Object Lock per un bucket, utilizzare uno dei seguenti metodi:

- Creare il bucket utilizzando Tenant Manager. Vedere "Crea bucket S3" .
- Crea il bucket utilizzando una richiesta CreateBucket con `x-amz-bucket-object-lock-enabled` intestazione della richiesta. Vedere "Operazioni sui bucket" .

S3 Object Lock richiede il controllo delle versioni dei bucket, che viene abilitato automaticamente al momento della creazione del bucket. Non è possibile sospendere il controllo delle versioni per il bucket. Vedere "Versionamento degli oggetti" .

Impostazioni di conservazione predefinite per un bucket

Quando S3 Object Lock è abilitato per un bucket, è possibile abilitare facoltativamente la conservazione predefinita per il bucket e specificare una modalità di conservazione predefinita e un periodo di conservazione predefinito.

Modalità di conservazione predefinita

- In modalità CONFORMITÀ:
  - L'oggetto non può essere eliminato finché non viene raggiunta la data di conservazione.
  - La data di conservazione dell'oggetto può essere aumentata, ma non diminuita.
  - La data di conservazione dell'oggetto non può essere rimossa finché non viene raggiunta tale data.
- In modalità GOVERNANCE:
  - Utenti con il `s3:BypassGovernanceRetention` il permesso può utilizzare il `x-amz-bypass-governance-retention: true` intestazione della richiesta per ignorare le impostazioni di conservazione.
  - Questi utenti possono eliminare una versione di un oggetto prima che venga raggiunta la data di conservazione.
  - Questi utenti possono aumentare, diminuire o rimuovere la data di conservazione di un oggetto.

Periodo di conservazione predefinito

Ogni bucket può avere un periodo di conservazione predefinito specificato in anni o giorni.

Come impostare la conservazione predefinita per un bucket

Per impostare la conservazione predefinita per un bucket, utilizzare uno dei seguenti metodi:

- Gestisci le impostazioni del bucket da Tenant Manager. Vedere"Crea un bucket S3" E"Aggiorna la conservazione predefinita del blocco degli oggetti S3" .
- Inviare una richiesta PutObjectLockConfiguration al bucket per specificare la modalità predefinita e il numero predefinito di giorni o anni.

PutObjectLockConfiguration

La richiesta PutObjectLockConfiguration consente di impostare e modificare la modalità di conservazione predefinita e il periodo di conservazione predefinito per un bucket in cui è abilitato S3 Object Lock. È anche possibile rimuovere le impostazioni di conservazione predefinite configurate in precedenza.

Quando nuove versioni di oggetti vengono acquisite nel bucket, viene applicata la modalità di conservazione predefinita se `x-amz-object-lock-mode` E `x-amz-object-lock-retain-until-date` non sono specificati. Il periodo di conservazione predefinito viene utilizzato per calcolare la data di conservazione fino a se `x-amz-object-lock-retain-until-date` non è specificato.

Se il periodo di conservazione predefinito viene modificato dopo l'acquisizione di una versione dell'oggetto, la data di conservazione fino alla versione dell'oggetto rimane invariata e non viene ricalcolata utilizzando il nuovo periodo di conservazione predefinito.

Devi avere il `s3:PutBucketObjectLockConfiguration` autorizzazione, oppure essere l'account root, per completare questa operazione.

IL `Content-MD5` l'intestazione della richiesta deve essere specificata nella richiesta PUT.

Richiedi esempio

Questo esempio abilita S3 Object Lock per un bucket e imposta la modalità di conservazione predefinita su CONFORMITÀ e il periodo di conservazione predefinito su 6 anni.

```
PUT /bucket?object-lock HTTP/1.1
Accept-Encoding: identity
Content-Length: 308
Host: host
Content-MD5: request header
User-Agent: s3sign/1.0.0 requests/2.24.0 python/3.8.2
X-Amz-Date: date
X-Amz-Content-SHA256: authorization-string
Authorization: authorization-string

<ObjectLockConfiguration>
    <ObjectLockEnabled>Enabled</ObjectLockEnabled>
    <Rule>
        <DefaultRetention>
            <Mode>COMPLIANCE</Mode>
            <Years>6</Years>
        </DefaultRetention>
    </Rule>
</ObjectLockConfiguration>
```

Come determinare la conservazione predefinita per un bucket

Per determinare se S3 Object Lock è abilitato per un bucket e per visualizzare la modalità di conservazione predefinita e il periodo di conservazione, utilizzare uno di questi metodi:

- Visualizza il bucket in Tenant Manager. Vedere "Visualizza i bucket S3" .
- Inviare una richiesta GetObjectLockConfiguration.

Ottieni configurazione blocco oggetto

La richiesta GetObjectLockConfiguration consente di determinare se S3 Object Lock è abilitato per un bucket e, in tal caso, di verificare se sono configurati una modalità di conservazione predefinita e un periodo di conservazione per il bucket.

Quando nuove versioni di oggetti vengono acquisite nel bucket, viene applicata la modalità di conservazione predefinita se `x-amz-object-lock-mode` non è specificato. Il periodo di conservazione predefinito viene utilizzato per calcolare la data di conservazione fino a se `x-amz-object-lock-retain-until-date` non è specificato.

Devi avere il `s3:GetBucketObjectLockConfiguration` autorizzazione, oppure essere l'account root, per completare questa operazione.

Richiedi esempio

```
GET /bucket?object-lock HTTP/1.1
Host: host
Accept-Encoding: identity
User-Agent: aws-cli/1.18.106 Python/3.8.2 Linux/4.4.0-18362-Microsoft
botocore/1.17.29
x-amz-date: date
x-amz-content-sha256: authorization-string
Authorization: authorization-string
```

Esempio di risposta

```
HTTP/1.1 200 OK
x-amz-id-2:
iVmcB7OXXJRkRH1FiVq1151/T24gRfpwpuZrEG11Bb9ImOMAAe98oxSpXlknabA0LTvBYJpSIX
k=
x-amz-request-id: B34E94CACB2CEF6D
Date: Fri, 04 Sep 2020 22:47:09 GMT
Transfer-Encoding: chunked
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<ObjectLockConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <ObjectLockEnabled>Enabled</ObjectLockEnabled>
    <Rule>
        <DefaultRetention>
            <Mode>COMPLIANCE</Mode>
            <Years>6</Years>
        </DefaultRetention>
    </Rule>
</ObjectLockConfiguration>
```

Come specificare le impostazioni di conservazione per un oggetto

Un bucket con S3 Object Lock abilitato può contenere una combinazione di oggetti con e senza impostazioni di conservazione S3 Object Lock.

Le impostazioni di conservazione a livello di oggetto vengono specificate tramite l'API REST S3. Le impostazioni di conservazione per un oggetto sostituiscono tutte le impostazioni di conservazione predefinite per il bucket.

Per ogni oggetto è possibile specificare le seguenti impostazioni:

- **Modalità di conservazione**: CONFORMITÀ o GOVERNANCE.
- **Retain-until-date**: data che specifica per quanto tempo la versione dell'oggetto deve essere conservata da StorageGRID.

    - In modalità CONFORMITÀ, se la data di conservazione è futura, l'oggetto può essere recuperato, ma non può essere modificato o eliminato. La data di conservazione può essere aumentata, ma questa data non può essere diminuita o rimossa.
    - In modalità GOVERNANCE, gli utenti con autorizzazione speciale possono ignorare l'impostazione di conservazione fino alla data indicata. Possono eliminare una versione di un oggetto prima che scada il periodo di conservazione. Possono anche aumentare, diminuire o addirittura rimuovere la data di conservazione.
- **Sospensione legale**: l'applicazione di una sospensione legale a una versione di un oggetto blocca immediatamente quell'oggetto. Ad esempio, potrebbe essere necessario applicare un blocco legale a un oggetto correlato a un'indagine o a una controversia legale. Una sospensione legale non ha una data di scadenza, ma rimane in vigore finché non viene rimossa esplicitamente.

  L'impostazione di conservazione legale per un oggetto è indipendente dalla modalità di conservazione e dalla data di conservazione fino alla data di scadenza. Se una versione di un oggetto è sottoposta a blocco legale, nessuno può eliminarla.

Per specificare le impostazioni di blocco degli oggetti S3 quando si aggiunge una versione dell'oggetto a un bucket, emettere un"MettiOggetto" ,"CopiaOggetto" , O"CreaCaricamentoMultiparte" richiesta.

Puoi usare quanto segue:

- `x-amz-object-lock-mode`, che può essere COMPLIANCE o GOVERNANCE (con distinzione tra maiuscole e minuscole).

  > Se specifichi `x-amz-object-lock-mode` , devi anche specificare `x-amz-object-lock-retain-until-date` .

- `x-amz-object-lock-retain-until-date`
    - Il valore retain-til-date deve essere nel formato `2020-08-10T21:46:00Z` . Sono consentite frazioni di secondo, ma vengono conservate solo 3 cifre decimali (precisione in millisecondi). Altri formati ISO 8601 non sono consentiti.
    - La data di conservazione deve essere futura.
- `x-amz-object-lock-legal-hold`

  Se la conservazione legale è ATTIVA (sensibile alle maiuscole e alle minuscole), l'oggetto viene sottoposto a conservazione legale. Se la conservazione legale è disattivata, non verrà applicata alcuna conservazione legale. Qualsiasi altro valore genera un errore 400 Bad Request (InvalidArgument).

Se si utilizza una di queste intestazioni di richiesta, tenere presente le seguenti restrizioni:

- IL `Content-MD5` l'intestazione della richiesta è obbligatoria se presente `x-amz-object-lock-*` l'intestazione della richiesta è presente nella richiesta PutObject. `Content-MD5` non è richiesto per CopyObject o CreateMultipartUpload.
- Se il bucket non ha S3 Object Lock abilitato e un `x-amz-object-lock-*` Se è presente l'intestazione della richiesta, viene restituito un errore 400 Bad Request (InvalidRequest).
- La richiesta PutObject supporta l'uso di `x-amz-storage-class: REDUCED_REDUNDANCY` per adattarsi al comportamento di AWS. Tuttavia, quando un oggetto viene inserito in un bucket con S3 Object Lock abilitato, StorageGRID eseguirà sempre un inserimento a doppio commit.

- Una successiva risposta alla versione GET o HeadObject includerà le intestazioni `x-amz-object-lock-mode` , `x-amz-object-lock-retain-until-date` , E `x-amz-object-lock-legal-hold` , se configurato e se il mittente della richiesta ha il corretto `s3:Get*` permessi.

Puoi usare il `s3:object-lock-remaining-retention-days` chiave di condizione della policy per limitare i periodi di conservazione minimi e massimi consentiti per i tuoi oggetti.

Come aggiornare le impostazioni di conservazione per un oggetto

Se è necessario aggiornare le impostazioni di conservazione o di blocco legale per una versione esistente di un oggetto, è possibile eseguire le seguenti operazioni sulle sottorisorse dell'oggetto:

- `PutObjectLegalHold`

  Se il nuovo valore di conservazione legale è impostato su ON, l'oggetto viene sottoposto a conservazione legale. Se il valore di sospensione legale è OFF, la sospensione legale viene revocata.

- `PutObjectRetention`
  - Il valore della modalità può essere COMPLIANCE o GOVERNANCE (con distinzione tra maiuscole e minuscole).
  - Il valore retain-til-date deve essere nel formato `2020-08-10T21:46:00Z` . Sono consentite frazioni di secondo, ma vengono conservate solo 3 cifre decimali (precisione in millisecondi). Altri formati ISO 8601 non sono consentiti.
  - Se una versione di un oggetto ha una data di conservazione (retain-til-date) esistente, è possibile solo aumentarla. Il nuovo valore deve essere nel futuro.

Come utilizzare la modalità GOVERNANCE

Gli utenti che hanno il `s3:BypassGovernanceRetention` l'autorizzazione può ignorare le impostazioni di conservazione attive di un oggetto che utilizza la modalità GOVERNANCE. Tutte le operazioni DELETE o PutObjectRetention devono includere `x-amz-bypass-governance-retention:true` intestazione della richiesta. Questi utenti possono eseguire le seguenti operazioni aggiuntive:

- Eseguire le operazioni DeleteObject o DeleteObjects per eliminare una versione dell'oggetto prima che scada il periodo di conservazione.

  Gli oggetti sottoposti a conservazione legale non possono essere eliminati. La conservazione legale deve essere DISATTIVATA.

- Eseguire operazioni PutObjectRetention che modificano la modalità di una versione dell'oggetto da GOVERNANCE a COMPLIANCE prima che sia trascorso il periodo di conservazione dell'oggetto.

  Non è mai consentito cambiare la modalità da COMPLIANCE a GOVERNANCE.

- Eseguire operazioni PutObjectRetention per aumentare, diminuire o rimuovere il periodo di conservazione di una versione dell'oggetto.

Informazioni correlate

- "Gestisci gli oggetti con S3 Object Lock"
- "Utilizzare S3 Object Lock per conservare gli oggetti"
- "Guida per l'utente di Amazon Simple Storage Service: blocco degli oggetti"

Crea la configurazione del ciclo di vita S3

È possibile creare una configurazione del ciclo di vita S3 per controllare quando oggetti specifici vengono eliminati dal sistema StorageGRID .

Il semplice esempio in questa sezione illustra come una configurazione del ciclo di vita S3 può controllare quando determinati oggetti vengono eliminati (scadono) da specifici bucket S3. L'esempio in questa sezione è solo a scopo illustrativo. Per i dettagli completi sulla creazione di configurazioni del ciclo di vita S3, vedere "Guida per l'utente di Amazon Simple Storage Service: gestione del ciclo di vita degli oggetti" . Si noti che StorageGRID supporta solo azioni di scadenza; non supporta azioni di transizione.

Qual è la configurazione del ciclo di vita?

Una configurazione del ciclo di vita è un insieme di regole applicate agli oggetti in bucket S3 specifici. Ogni regola specifica quali oggetti sono interessati e quando tali oggetti scadranno (in una data specifica o dopo un certo numero di giorni).

StorageGRID supporta fino a 1.000 regole del ciclo di vita in una configurazione del ciclo di vita. Ogni regola può includere i seguenti elementi XML:

- Scadenza: elimina un oggetto quando viene raggiunta una data specificata o quando viene raggiunto un numero di giorni specificato, a partire dal momento in cui l'oggetto è stato acquisito.
- NoncurrentVersionExpiration: elimina un oggetto quando viene raggiunto un numero di giorni specificato, a partire dal momento in cui l'oggetto è diventato non corrente.
- Filtro (Prefisso, Tag)
- Stato
- ID

Ogni oggetto segue le impostazioni di conservazione del ciclo di vita di un bucket S3 o di una policy ILM. Quando viene configurato un ciclo di vita del bucket S3, le azioni di scadenza del ciclo di vita sovrascrivono il criterio ILM per gli oggetti che corrispondono al filtro del ciclo di vita del bucket. Gli oggetti che non corrispondono al filtro del ciclo di vita del bucket utilizzano le impostazioni di conservazione del criterio ILM. Se un oggetto corrisponde a un filtro del ciclo di vita del bucket e non vengono specificate esplicitamente azioni di scadenza, le impostazioni di conservazione del criterio ILM non vengono utilizzate e si implica che le versioni dell'oggetto vengano conservate per sempre. Vedere "Esempio di priorità per il ciclo di vita del bucket S3 e la policy ILM" .

Di conseguenza, un oggetto potrebbe essere rimosso dalla griglia anche se le istruzioni di posizionamento in una regola ILM sono ancora valide per l'oggetto. Oppure, un oggetto potrebbe essere mantenuto sulla griglia anche dopo che tutte le istruzioni di posizionamento ILM per l'oggetto sono scadute. Per maggiori dettagli, vedere "Come funziona l'ILM durante la vita di un oggetto" .

> La configurazione del ciclo di vita del bucket può essere utilizzata con bucket in cui è abilitato S3 Object Lock, ma la configurazione del ciclo di vita del bucket non è supportata per i bucket Compliant legacy.

StorageGRID supporta l'utilizzo delle seguenti operazioni bucket per gestire le configurazioni del ciclo di vita:

- DeleteBucketLifecycle
- GetBucketLifecycleConfiguration

- Configurazione del ciclo di vita di PutBucket

Crea la configurazione del ciclo di vita

Come primo passaggio nella creazione di una configurazione del ciclo di vita, si crea un file JSON che include una o più regole. Ad esempio, questo file JSON include tre regole, come segue:

1. La regola 1 si applica solo agli oggetti che corrispondono al prefisso `category1` / e che hanno un `key2` valore di `tag2` . IL `Expiration` Il parametro specifica che gli oggetti che corrispondono al filtro scadranno a mezzanotte del 22 agosto 2020.
2. La regola 2 si applica solo agli oggetti che corrispondono al prefisso `category2` /. IL `Expiration` Il parametro specifica che gli oggetti che corrispondono al filtro scadranno 100 giorni dopo essere stati acquisiti.

   > Le regole che specificano un numero di giorni sono relative al momento in cui l'oggetto è stato ingerito. Se la data corrente supera la data di acquisizione più il numero di giorni, alcuni oggetti potrebbero essere rimossi dal bucket non appena viene applicata la configurazione del ciclo di vita.

3. La regola 3 si applica solo agli oggetti che corrispondono al prefisso `category3` /. IL `Expiration` Il parametro specifica che tutte le versioni non correnti degli oggetti corrispondenti scadranno 50 giorni dopo essere diventate non correnti.

```
{
    "Rules": [
        {
            "ID": "rule1",
            "Filter": {
                "And": {
                    "Prefix": "category1/",
                    "Tags": [
                        {
                            "Key": "key2",
                            "Value": "tag2"
                        }
                    ]
                }
            },
            "Expiration": {
                "Date": "2020-08-22T00:00:00Z"
            },
            "Status": "Enabled"
        },
        {
            "ID": "rule2",
            "Filter": {
                "Prefix": "category2/"
            },
            "Expiration": {
                "Days": 100
            },
            "Status": "Enabled"
        },
        {
            "ID": "rule3",
            "Filter": {
                "Prefix": "category3/"
            },
            "NoncurrentVersionExpiration": {
                "NoncurrentDays": 50
            },
            "Status": "Enabled"
        }
    ]
}
```

Applica la configurazione del ciclo di vita al bucket

Dopo aver creato il file di configurazione del ciclo di vita, puoi applicarlo a un bucket inviando una richiesta PutBucketLifecycleConfiguration.

Questa richiesta applica la configurazione del ciclo di vita nel file di esempio agli oggetti in un bucket denominato `testbucket` .

```
aws s3api --endpoint-url <StorageGRID endpoint> put-bucket-lifecycle-
configuration
--bucket testbucket --lifecycle-configuration file://bktjson.json
```

Per convalidare che una configurazione del ciclo di vita sia stata applicata correttamente al bucket, inviare una richiesta GetBucketLifecycleConfiguration. Per esempio:

```
aws s3api --endpoint-url <StorageGRID endpoint> get-bucket-lifecycle-
configuration
 --bucket testbucket
```

Una risposta positiva elenca la configurazione del ciclo di vita appena applicata.

Convalida che la scadenza del ciclo di vita del bucket si applichi all'oggetto

È possibile determinare se una regola di scadenza nella configurazione del ciclo di vita si applica a un oggetto specifico quando si invia una richiesta PutObject, HeadObject o GetObject. Se si applica una regola, la risposta include un `Expiration` parametro che indica quando scade l'oggetto e quale regola di scadenza è stata rispettata.

> Poiché il ciclo di vita del bucket sostituisce ILM, `expiry-date` viene mostrata la data effettiva in cui l'oggetto verrà eliminato. Per maggiori dettagli, vedere "Come viene determinata la ritenzione dell'oggetto" .

Ad esempio, questa richiesta PutObject è stata emessa il 22 giugno 2020 e inserisce un oggetto nel `testbucket` secchio.

```
aws s3api --endpoint-url <StorageGRID endpoint> put-object
--bucket testbucket --key obj2test2 --body bktjson.json
```

La risposta di successo indica che l'oggetto scadrà tra 100 giorni (01 ottobre 2020) e che corrisponde alla Regola 2 della configurazione del ciclo di vita.

```
{
      *"Expiration": "expiry-date=\"Thu, 01 Oct 2020 09:07:49 GMT\", rule-
id=\"rule2\"",
      "ETag": "\"9762f8a803bc34f5340579d4446076f7\""
}
```

Ad esempio, questa richiesta HeadObject è stata utilizzata per ottenere metadati per lo stesso oggetto nel bucket testbucket.

```
aws s3api --endpoint-url <StorageGRID endpoint> head-object
--bucket testbucket --key obj2test2
```

La risposta di successo include i metadati dell'oggetto e indica che l'oggetto scadrà tra 100 giorni e che soddisfa la Regola 2.

```
{
      "AcceptRanges": "bytes",
      *"Expiration": "expiry-date=\"Thu, 01 Oct 2020 09:07:48 GMT\", rule-
id=\"rule2\"",
      "LastModified": "2020-06-23T09:07:48+00:00",
      "ContentLength": 921,
      "ETag": "\"9762f8a803bc34f5340579d4446076f7\""
      "ContentType": "binary/octet-stream",
      "Metadata": {}
}
```

Per i bucket abilitati al controllo delle versioni, `x-amz-expiration` l'intestazione di risposta si applica solo alle versioni correnti degli oggetti.

Raccomandazioni per l'implementazione dell'API REST S3

Quando si implementa l'API REST S3 per l'uso con StorageGRID, è necessario seguire queste raccomandazioni.

Raccomandazioni per HEAD su oggetti inesistenti

Se la tua applicazione controlla regolarmente se un oggetto esiste in un percorso in cui non ti aspetti che l'oggetto esista effettivamente, dovresti usare "Disponibile""coerenza" . Ad esempio, dovresti usare la coerenza "Disponibile" se la tua applicazione esegue l'HEAD di una posizione prima di eseguirvi un PUT.

In caso contrario, se l'operazione HEAD non trova l'oggetto, è possibile che venga visualizzato un numero elevato di errori 500 Internal Server se due o più nodi di archiviazione nello stesso sito non sono disponibili o un sito remoto non è raggiungibile.

È possibile impostare la coerenza "Disponibile" per ogni bucket utilizzando"PUT Consistenza del secchio"

richiesta oppure è possibile specificare la coerenza nell'intestazione della richiesta per una singola operazione API.

Raccomandazioni per le chiavi degli oggetti

Seguire questi consigli per i nomi delle chiavi degli oggetti, in base al momento in cui il bucket è stato creato per la prima volta.

Bucket creati in StorageGRID 11.4 o versioni precedenti

- Non utilizzare valori casuali come primi quattro caratteri delle chiavi degli oggetti. Ciò è in contrasto con la precedente raccomandazione di AWS per i prefissi delle chiavi. Utilizzare invece prefissi non casuali e non univoci, come ad esempio `image` .
- Se si segue la precedente raccomandazione di AWS di utilizzare caratteri casuali e univoci nei prefissi delle chiavi, aggiungere un nome di directory come prefisso alle chiavi degli oggetti. Cioè, usa questo formato:

  `mybucket/mydir/f8e3-image3132.jpg`

  Invece di questo formato:

  `mybucket/f8e3-image3132.jpg`

Bucket creati in StorageGRID 11.4 o versioni successive

Non è necessario limitare i nomi delle chiavi degli oggetti per soddisfare le migliori pratiche in termini di prestazioni. Nella maggior parte dei casi, è possibile utilizzare valori casuali per i primi quattro caratteri dei nomi delle chiavi degli oggetti.

> Un'eccezione è il carico di lavoro S3 che rimuove continuamente tutti gli oggetti dopo un breve periodo di tempo. Per ridurre al minimo l'impatto sulle prestazioni in questo caso d'uso, modificare una parte iniziale del nome della chiave ogni diverse migliaia di oggetti con qualcosa come la data. Ad esempio, supponiamo che un client S3 scriva in genere 2.000 oggetti al secondo e che la policy ILM o del ciclo di vita del bucket rimuova tutti gli oggetti dopo tre giorni. Per ridurre al minimo l'impatto sulle prestazioni, potresti denominare le chiavi utilizzando uno schema come questo: `/mybucket/mydir/`*`yyyymmddhhmmss-random_UUID`*`.jpg`

Consigli per le "letture di intervallo"

Se il"opzione globale per comprimere gli oggetti memorizzati" è abilitato, le applicazioni client S3 dovrebbero evitare di eseguire operazioni GetObject che specificano un intervallo di byte da restituire. Queste operazioni di "lettura di intervallo" sono inefficienti perché StorageGRID deve effettivamente decomprimere gli oggetti per accedere ai byte richiesti. Le operazioni GetObject che richiedono un intervallo ridotto di byte da un oggetto molto grande sono particolarmente inefficienti; ad esempio, non è efficiente leggere un intervallo di 10 MB da un oggetto compresso da 50 GB.

Se gli intervalli vengono letti da oggetti compressi, le richieste del client potrebbero scadere.

> Se è necessario comprimere oggetti e l'applicazione client deve utilizzare letture di intervallo, aumentare il timeout di lettura per l'applicazione.

Informazioni sul copyright

Informazioni sul marchio commerciale