Come StorageGRID implementa l'API REST S3

StorageGRID software

NetApp
July 23, 2026

Sommario

Come StorageGRID implementa l'API REST S3

Come StorageGRID S3 REST API risolve le richieste client 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 su quando il sistema StorageGRID completa una determinata richiesta e non su quando i client S3 iniziano un'operazione.

Come StorageGRID S3 REST API bilancia disponibilità e coerenza

La coerenza garantisce un equilibrio tra la disponibilità degli oggetti e la coerenza di quegli oggetti tra i diversi Storage Node e siti. Puoi modificare la coerenza come richiesto dalla tua applicazione.

Per impostazione predefinita, StorageGRID garantisce la coerenza di lettura dopo la scrittura per gli oggetti appena creati. Qualsiasi operazione GET successiva a una PUT completata con successo potrà leggere i dati appena scritti. Le sovrascritture di oggetti esistenti, gli aggiornamenti dei metadati e le eliminazioni sono eventualmente coerenti.

Se vuoi eseguire operazioni sugli oggetti con una consistenza diversa, puoi:

- Specifica una coerenza per ogni bucket.
- Specifica una coerenza per ogni operazione API.
- Modifica la coerenza predefinita dell'intera griglia eseguendo una delle seguenti operazioni:
  - In Grid Manager, vai a **Configurazione** > **Sistema** > **Impostazioni di archiviazione** > **Coerenza predefinita del bucket**.
  - .

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

Valori di coerenza

La coerenza influisce su come i metadati che StorageGRID utilizza per tracciare gli oggetti vengono distribuiti tra i nodi. La coerenza influisce sulla disponibilità degli oggetti per le richieste dei client.

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

- **Tutti**: Tutti i nodi ricevono immediatamente i metadati dell'oggetto oppure la richiesta fallirà.
- **Strong-global**: Garantisce la coerenza read-after-write per tutte le richieste client su tutti i siti. Quando la semantica del quorum è configurata, si applicano i seguenti comportamenti:

- Consente la tolleranza ai guasti del sito per le richieste dei client quando le griglie hanno tre o più siti. Le griglie con due siti non avranno la tolleranza ai guasti del sito.
- Le seguenti operazioni S3 non andranno a buon fine se un sito non è disponibile:
  - DeleteBucketEncryption
  - PutBucketBranch
  - PutBucketEncryption
  - PutBucketVersioning
  - PutObjectLegalHold
  - PutObjectLockConfiguration
  - PutObjectRetention

Se necessario, puoi "Configura la semantica del quorum di StorageGRID per una strong-global consistency".

- **Strong-site**: i metadati degli oggetti vengono distribuiti immediatamente agli altri nodi del sito. Garantisce la coerenza read-after-write per tutte le richieste dei client all'interno di un sito.
- **Read-after-new-write**: Garantisce la coerenza read-after-write per i nuovi oggetti e la coerenza finale per gli aggiornamenti degli oggetti. Offre elevata disponibilità e garanzie di 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, usalo 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 che non esistono). Non supportato per i bucket S3 FabricPool.

Usa la coerenza "Read-after-new-write" e "Available"

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 consistenza bassa.
- Se la ricerca fallisce, la ripeti al valore di coerenza successivo finché non raggiungi una coerenza equivalente al comportamento per 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 più copie dei metadati dell'oggetto siano disponibili in ogni sito, puoi ricevere un numero elevato di errori 500 Internal Server se due o più Storage Node nello stesso sito non sono disponibili.

A meno che tu non abbia bisogno di garanzie di coerenza simili a quelle di Amazon S3, puoi 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 coerenza eventuale. Non riprova un'operazione non riuscita con coerenza crescente, 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 e devi specificare la coerenza nell'intestazione della richiesta. Questo esempio imposta la coerenza su "Strong-site" per un'operazione GetObject.

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

Devi usare la stessa coerenza sia per le operazioni PutObject che GetObject.

Specifica la coerenza per il bucket

Per impostare la coerenza per un bucket, puoi usare la richiesta "Coerenza PUT Bucket" StorageGRID. Oppure puoi farlo "cambiare la consistenza di un bucket" dal Tenant Manager.

Quando imposti la consistenza di un bucket, tieni 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 per una singola operazione API ha la precedenza sulla coerenza per il bucket.
- In generale, i bucket dovrebbero utilizzare la coerenza predefinita, "Read-after-new-write". Se le richieste non funzionano correttamente, modifica il comportamento del client dell'applicazione se possibile. Oppure configura il client per specificare la coerenza per ogni richiesta API. Imposta la coerenza a livello di bucket solo come ultima risorsa.

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

Sia la scelta del livello di coerenza che la regola ILM influiscono sulla modalità di protezione degli oggetti. Queste impostazioni possono interagire.

Ad esempio, il livello di coerenza utilizzato durante la memorizzazione di un oggetto 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 che ai dati di un oggetto per soddisfare le richieste dei client, la selezione di livelli di protezione corrispondenti per la coerenza e il comportamento di acquisizione può offrire una migliore protezione iniziale dei dati e risposte di sistema più prevedibili.

Le seguenti "opzioni di ingest" sono disponibili per le regole ILM:

Commit doppio

StorageGRID crea immediatamente copie provvisorie dell'oggetto e restituisce successo al client. Le copie specificate nella regola ILM vengono create quando possibile.

Rigoroso

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

Equilibrato

StorageGRID tenta di creare tutte le copie specificate nella regola ILM al momento dell'acquisizione; se ciò non è possibile, vengono create copie intermedie e il client riceve una conferma di successo. Le copie

specificate nella regola ILM vengono create quando possibile.

Esempio di come la coerenza e la regola ILM possono interagire

Supponiamo che tu abbia una griglia a tre siti con la seguente regola ILM e la seguente coerenza:

- **Regola ILM**: Crea tre copie dell'oggetto, una nel sito locale e una in ciascun sito remoto. Usa il comportamento di acquisizione rigoroso.
- **Coerenza**: Forte-globale (i metadati degli oggetti vengono distribuiti immediatamente a più siti).

Quando un client memorizza un oggetto nella griglia, StorageGRID crea tutte e tre le copie dell'oggetto e distribuisce i metadati a più siti prima di restituire conferma al client.

L'oggetto è completamente protetto dalla perdita al momento della ricezione del messaggio di avvenuta acquisizione. Ad esempio, se il sito locale viene perso poco dopo l'acquisizione, copie sia dei dati dell'oggetto che dei metadati dell'oggetto esistono comunque nei siti remoti. L'oggetto è completamente recuperabile dagli altri siti.

Se invece usi la stessa regola ILM e la coerenza strong-site, il client potrebbe ricevere un messaggio di successo dopo che i dati dell'oggetto sono stati replicati nei siti remoti, ma prima che i metadati dell'oggetto vengano distribuiti lì. In questo caso, il livello di protezione dei metadati dell'oggetto non corrisponde al livello di protezione dei dati dell'oggetto. Se il sito locale viene perso poco dopo l'ingestione, i metadati dell'oggetto vengono persi. L'oggetto non può essere recuperato.

La relazione tra coerenza e regole ILM può essere complessa. Contatta NetApp se hai bisogno di assistenza.

Come StorageGRID utilizza il versioning degli oggetti

Puoi impostare lo stato di versioning di un bucket se vuoi conservare più versioni di ciascun oggetto. Abilitare il versioning per un bucket può aiutarti a proteggere dalla cancellazione accidentale degli oggetti e ti permette di recuperare e ripristinare le versioni precedenti di un oggetto.

Il sistema StorageGRID implementa il versioning con supporto per la maggior parte delle funzionalità, seppur con alcune limitazioni. StorageGRID supporta fino a 10.000 versioni per ciascun oggetto.

Il versioning degli oggetti può essere combinato con StorageGRID information lifecycle management (ILM) o con la configurazione del ciclo di vita dei bucket S3. Devi abilitare esplicitamente il versioning per ogni bucket. Quando il versioning è abilitato per un bucket, a ogni oggetto aggiunto al bucket viene assegnato un ID di versione, generato dal sistema StorageGRID.

L'uso di MFA (autenticazione a più fattori) Delete non è supportato.

> La gestione delle versioni può essere abilitata solo sui bucket creati con StorageGRID versione 10.3 o successiva.

ILM e versioning

Le policy ILM vengono applicate a ogni versione di un oggetto. Un processo di scansione ILM analizza continuamente tutti gli oggetti e li rivaluta rispetto alla policy ILM corrente. Qualsiasi modifica apportata alle policy ILM viene applicata a tutti gli oggetti precedentemente acquisiti. Questo include le versioni precedentemente acquisite se la gestione delle versioni è abilitata. La scansione ILM applica le nuove

modifiche ILM agli oggetti precedentemente acquisiti.

Per gli oggetti S3 nei bucket con versioning abilitato, il supporto per il versioning ti permette di creare regole ILM che usano "Tempo non corrente" come tempo di riferimento (seleziona **Sì** alla domanda "Applica 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 correnti. Usando un filtro "Tempo non corrente" puoi creare criteri che riducono l'impatto sullo spazio di archiviazione delle versioni precedenti degli oggetti.

> Quando carichi una nuova versione di un oggetto usando un'operazione di caricamento in più parti, l'ora non corrente della versione originale dell'oggetto riflette il momento in cui è stato creato il caricamento in più parti per la nuova versione, non quando il caricamento in più parti è stato completato. In casi limitati, l'ora non corrente della versione originale potrebbe essere di ore o giorni precedente rispetto all'ora della versione corrente.

Informazioni correlate

- "Come vengono eliminati gli oggetti versionati di S3"
- "Regole e politiche ILM per gli oggetti versionati S3 (Esempio 4)".

Usa l'API REST S3 per configurare StorageGRID S3 Object Lock

Se l'impostazione globale S3 Object Lock è abilitata per il tuo sistema StorageGRID, puoi creare bucket con S3 Object Lock attivo. Puoi specificare il periodo di conservazione predefinito 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 tuo sistema StorageGRID, puoi scegliere di abilitare S3 Object Lock quando crei ogni bucket.

Il blocco degli oggetti S3 è un'impostazione permanente che puoi abilitare solo quando crei un bucket. Non puoi aggiungere o disabilitare il blocco degli oggetti S3 dopo che un bucket è stato creato.

Per abilitare il blocco degli oggetti S3 per un bucket, usa uno di questi metodi:

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

S3 Object Lock richiede il versioning del bucket, che viene abilitato automaticamente quando il bucket viene creato. Non puoi sospendere il versioning per il bucket. Vedi "Versioning degli oggetti".

Impostazioni di conservazione predefinite per un bucket

Quando il blocco degli oggetti S3 è abilitato per un bucket, puoi facoltativamente abilitare 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 COMPLIANCE:
  - L'oggetto non può essere eliminato fino al raggiungimento della sua retain-until-date.
  - La data di conservazione dell'oggetto può essere aumentata, ma non può essere diminuita.
  - La data di conservazione dell'oggetto non può essere rimossa fino al raggiungimento di tale data.
- In modalità GOVERNANCE:
  - Gli utenti con l' `s3:BypassGovernanceRetention`autorizzazione possono usare l' `x-amz-bypass-governance-retention: true`intestazione della richiesta per bypassare le impostazioni di conservazione.
  - Questi utenti possono eliminare una versione di un oggetto prima che venga raggiunta la data di conservazione prevista.
  - Questi utenti possono aumentare, diminuire o rimuovere la retain-until-date di un oggetto.

Periodo di conservazione predefinito

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

Come impostare la conservazione predefinita per un bucket

Per impostare il periodo di conservazione predefinito per un bucket, usa uno di questi metodi:

- Gestisci le impostazioni del bucket dal Tenant Manager. Vedi "Crea un bucket S3" e "Aggiorna la conservazione predefinita del blocco oggetto S3".
- Invia una richiesta PutObjectLockConfiguration per il bucket per specificare la modalità predefinita e il numero predefinito di giorni o anni.

PutObjectLockConfiguration

La richiesta PutObjectLockConfiguration ti permette di impostare e modificare la modalità di conservazione predefinita e il periodo di conservazione predefinito per un bucket che ha S3 Object Lock abilitato. Puoi anche rimuovere le impostazioni di conservazione predefinite configurate in precedenza.

Quando nuove versioni degli oggetti vengono inserite 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 retain-until-date 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 di un oggetto, la retain-until-date della versione dell'oggetto rimane invariata e non viene ricalcolata utilizzando il nuovo periodo di conservazione predefinito.

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

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

Esempio di richiesta

Questo esempio abilita S3 Object Lock per un bucket e imposta la modalità di conservazione predefinita su COMPLIANCE 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 il periodo di conservazione predefinito 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, usa uno di questi metodi:

- Visualizza il bucket in Tenant Manager. Vedi "Visualizza i bucket S3".
- Invia una richiesta GetObjectLockConfiguration.

GetObjectLockConfiguration

La richiesta GetObjectLockConfiguration ti permette di verificare se S3 Object Lock è abilitato per un bucket e, se lo è, vedere se sono configurati una modalità di conservazione predefinita e un periodo di conservazione per il bucket.

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

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

Esempio di richiesta

```
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 di S3. Le impostazioni di conservazione per un oggetto sovrascrivono qualsiasi impostazione di conservazione predefinita per il bucket.

Puoi specificare le seguenti impostazioni per ogni oggetto:

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

- In modalità COMPLIANCE, se la retain-until-date è futura, l'oggetto può essere recuperato, ma non può essere modificato o eliminato. La retain-until-date può essere aumentata, ma questa data non può essere diminuita o rimossa.
- In modalità GOVERNANCE, gli utenti con autorizzazioni speciali possono ignorare l'impostazione della data di conservazione. Possono eliminare una versione di un oggetto prima che il periodo di conservazione sia trascorso. Possono anche aumentare, diminuire o persino rimuovere la data di conservazione.

- **Blocco legale**: L'applicazione di un blocco legale a una versione di un oggetto blocca immediatamente quell'oggetto. Ad esempio, potresti dover applicare un blocco legale a un oggetto correlato a un'indagine o a una controversia legale. Un blocco legale non ha una data di scadenza, ma rimane in vigore finché non viene esplicitamente rimosso.

  L'impostazione di blocco legale per un oggetto è indipendente dalla modalità di conservazione e dalla data di conservazione fino alla quale l'oggetto verrà conservato. Se una versione di un oggetto è soggetta a blocco legale, nessuno può eliminare quella versione.

Per specificare le impostazioni di S3 Object Lock quando aggiungi una versione di oggetto a un bucket, invia una richiesta "PutObject", "CopyObject" o "CreateMultipartUpload".

Puoi usare quanto segue:

- `x-amz-object-lock-mode`, che può essere COMPLIANCE o GOVERNANCE (sensibile alle maiuscole/minuscole).

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

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

  Se il legal hold è ON (case-sensitive), l'oggetto viene sottoposto a legal hold. Se il legal hold è OFF, non viene applicato alcun legal hold. Qualsiasi altro valore genera un errore 400 Bad Request (InvalidArgument).

Se utilizzi una qualsiasi di queste intestazioni di richiesta, tieni presente queste restrizioni:

- L' `Content-MD5`intestazione della richiesta è obbligatoria se è presente una qualsiasi `x-amz-object-lock-*`intestazione della richiesta nella richiesta PutObject. `Content-MD5`Non è obbligatoria per CopyObject o CreateMultipartUpload.
- Se il bucket non ha S3 Object Lock abilitato e una `x-amz-object-lock-*`intestazione di richiesta è presente, viene restituito un errore 400 Bad Request (InvalidRequest).
- La richiesta PutObject supporta l'uso di `x-amz-storage-class: REDUCED_REDUNDANCY` per eguagliare il comportamento di AWS. Tuttavia, quando un oggetto viene inserito in un bucket con S3 Object Lock abilitato, StorageGRID eseguirà sempre un ingest dual-commit.
- Una successiva risposta GET o HeadObject di versione includerà le intestazioni `x-amz-object-lock-`

`mode`, `x-amz-object-lock-retain-until-date` e `x-amz-object-lock-legal-hold`, se configurate e se chi invia la richiesta ha le corrette `s3:Get*` autorizzazioni.

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

Come aggiornare le impostazioni di conservazione per un oggetto

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

- `PutObjectLegalHold`

  Se il nuovo valore di legal hold è ON, l'oggetto viene sottoposto a legal hold. Se il valore di legal hold è OFF, il legal hold viene rimosso.

- `PutObjectRetention`
  - Il valore della modalità può essere COMPLIANCE o GOVERNANCE (case sensitive).
  - Il valore retain-until-date deve essere nel formato `2020-08-10T21:46:00Z`. Sono ammessi i secondi frazionari, ma vengono conservate solo 3 cifre decimali (precisione in millisecondi). Altri formati ISO 8601 non sono ammessi.
  - Se a una versione di un oggetto è già associata una data di conservazione (retain-until-date), puoi solo aumentarla. Il nuovo valore deve essere nel futuro.

Come usare la modalità GOVERNANCE

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

- Esegui le operazioni DeleteObject o DeleteObjects per eliminare una versione di un oggetto prima che il suo periodo di conservazione sia trascorso.

  Gli oggetti soggetti a blocco legale non possono essere eliminati. Il blocco legale deve essere disattivato.

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

  Il passaggio dalla modalità COMPLIANCE alla modalità GOVERNANCE non è mai consentito.

- Esegui 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"
- "Usa S3 Object Lock per conservare gli oggetti"
- "Guida utente di Amazon Simple Storage Service: Blocco degli oggetti"

Scopri la configurazione del ciclo di vita S3 per StorageGRID

Puoi creare una configurazione del ciclo di vita S3 per controllare quando determinati oggetti 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 dettagli completi sulla creazione di configurazioni del ciclo di vita S3, vedi "Guida per l'utente di Amazon Simple Storage Service: gestione del ciclo di vita degli oggetti". Nota che StorageGRID supporta solo le azioni di scadenza; non supporta le azioni di transizione.

Che cos'è la configurazione del ciclo di vita

Una configurazione del ciclo di vita è un insieme di regole applicate agli oggetti in specifici bucket S3. Ogni regola specifica quali oggetti sono interessati e quando questi 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 un file di 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 da quando l'oggetto è stato acquisito.
- NoncurrentVersionExpiration: Elimina un oggetto al raggiungimento di 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 di un bucket S3, le azioni di scadenza del ciclo di vita hanno la precedenza sulla policy 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 della policy 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 della policy ILM non vengono utilizzate e si intende che le versioni degli oggetti vengano conservate per sempre. Vedi "Esempi 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 rimanere sulla griglia anche dopo la scadenza di tutte le istruzioni di posizionamento ILM per l'oggetto. Per ulteriori dettagli, vedi "Come funziona ILM durante l'intero ciclo di vita di un oggetto".

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

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

- DeleteBucketLifecycle

- GetBucketLifecycleConfiguration
- PutBucketLifecycleConfiguration

Crea configurazione del ciclo di vita

Come primo passo nella creazione di una configurazione del ciclo di vita, crei 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` 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` parametro specifica che gli oggetti che corrispondono al filtro scadranno 100 giorni dopo essere stati ingeriti.

   > Le regole che specificano un numero di giorni sono relative al momento in cui l'oggetto è stato acquisito. 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` 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, lo applichi 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 verificare che una configurazione del ciclo di vita sia stata applicata correttamente al bucket, invia una richiesta GetBucketLifecycleConfiguration. Ad esempio:

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

Una risposta positiva elenca la configurazione del ciclo di vita che hai appena applicato.

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

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

> Poiché il ciclo di vita del bucket ha la precedenza su ILM, la `expiry-date` data mostrata è la data effettiva in cui l'oggetto verrà eliminato. Per dettagli, vedi "Come viene determinata la retention dell'oggetto".

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

```
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 ott 2020) e che corrispondeva 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 i 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 ha soddisfatto 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 con versioning abilitato, l' `x-amz-expiration`intestazione della risposta si applica solo alle versioni correnti degli oggetti.

Raccomandazioni per l'implementazione dell'API REST S3 con StorageGRID

Dovresti seguire queste raccomandazioni quando implementi l'API REST S3 per l'utilizzo con StorageGRID.

Raccomandazioni per HEAD su oggetti inesistenti

Se la tua applicazione verifica regolarmente se un oggetto esiste in un percorso in cui non ti aspetti che l'oggetto esista effettivamente, dovresti usare la consistenza "Available" "coerenza". Ad esempio, dovresti usare la consistenza "Available" se la tua applicazione esegue HEAD su una posizione prima di scriverci tramite PUT.

Altrimenti, se l'operazione HEAD non trova l'oggetto, potresti ricevere un numero elevato di errori 500 Internal Server Error se due o più Storage Node nello stesso sito non sono disponibili o se un remote site non è

raggiungibile.

Puoi impostare la coerenza "Available" per ciascun bucket usando la richiesta "Coerenza PUT Bucket", oppure puoi specificare la coerenza nell'intestazione della richiesta per una singola operazione API.

Raccomandazioni per le chiavi degli oggetti

Segui questi suggerimenti per i nomi delle chiavi degli oggetti, in base a quando 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. Utilizza invece prefissi non casuali e non univoci, come `image`.
- Se segui la precedente raccomandazione di AWS di utilizzare caratteri casuali e univoci nei prefissi delle chiavi, anteponi il nome della directory alle chiavi degli oggetti. Ovvero, utilizza 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 rispettare le best practice in termini di prestazioni. Nella maggior parte dei casi, puoi usare valori casuali per i primi quattro caratteri dei nomi delle chiavi degli oggetti.

> Un'eccezione a questa regola è rappresentata da un 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, varia una parte iniziale del nome della chiave ogni alcune migliaia di oggetti, ad esempio usando la data. Ad esempio, supponi che un client S3 scriva normalmente 2.000 oggetti al secondo e che la policy ILM o la policy di ciclo di vita del bucket rimuova tutti gli oggetti dopo tre giorni. Per ridurre al minimo l'impatto sulle prestazioni, potresti assegnare i nomi alle chiavi seguendo uno schema come questo: `/mybucket/mydir/yyyymmddhhmmss-random_UUID.jpg`

Consigli per le "range reads"

Se la "${post_edited_translations.segment}" è abilitata, 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 piccolo intervallo di byte da un oggetto molto grande sono particolarmente inefficienti; ad esempio, è inefficiente leggere un intervallo di 10 MB da un oggetto compresso di 50 GB.

${post_edited_translations.segment}

> ${post_edited_translations.segment}

Informazioni sul copyright

Informazioni sul marchio commerciale