Come StorageGRID implementa l'API REST S3

StorageGRID 11.9

NetApp
November 08, 2024

Sommario

Come StorageGRID implementa l'API REST S3

Richieste client in conflitto

Le richieste dei client in conflitto, come due client che scrivono sulla stessa chiave, vengono risolte in base alle "ultime vincite".

La tempistica per la valutazione degli "ultimi successi" si basa sul momento in cui il sistema StorageGRID completa una data richiesta e non sul momento in cui i client S3 iniziano un'operazione.

Valori di coerenza

La coerenza fornisce un equilibrio tra la disponibilità degli oggetti e la loro coerenza in diversi nodi e siti storage. È possibile modificare la coerenza come richiesto dall'applicazione.

Per impostazione predefinita, StorageGRID garantisce la coerenza di lettura dopo scrittura per gli oggetti appena creati. Qualsiasi GET che segue UN PUT completato con successo sarà in grado di leggere i dati appena scritti. Le sovrascritture degli oggetti esistenti, gli aggiornamenti dei metadati e le eliminazioni sono coerenti. Le sovrascritture in genere richiedono secondi o minuti per la propagazione, ma possono richiedere fino a 15 giorni.

Se si desidera eseguire operazioni a oggetti con una coerenza diversa, è possibile:

- Specificare una coerenza per ogni secchio.
- Specificare una coerenza per Ogni operazione API.
- Modificare la coerenza predefinita a livello di griglia eseguendo una delle seguenti operazioni:
  - In Grid Manager, andare a **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, vedere il registro di controllo situato in `/var/local/log` (cercare **consistencyLevel**).

Valori di coerenza

La coerenza influisce sul modo in cui i metadati utilizzati da StorageGRID per tenere traccia degli 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:

- **All**: Tutti i nodi ricevono i dati immediatamente, oppure la richiesta non riesce.
- **Strong-Global**: Garantisce la coerenza di lettura dopo scrittura per tutte le richieste dei client in tutti i siti.
- **Strong-Site**: Garantisce la coerenza di lettura dopo scrittura per tutte le richieste dei client all'interno di un sito.
- **Read-after-new-write**: (Default) fornisce coerenza lettura dopo scrittura per nuovi oggetti ed eventuale

coerenza per gli aggiornamenti degli oggetti. Offre alta disponibilità e garanzie di protezione dei dati. Consigliato per la maggior parte dei casi.

- **Available**: Fornisce una 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 raramente letti o per operazioni HEAD o GET su chiavi che non esistono). Non supportato per i bucket S3 FabricPool.

Utilizzare 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 bassa coerenza.
- Se la ricerca non riesce, ripete la ricerca al valore di coerenza successivo finché non raggiunge 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 degli oggetti raggiungerà sempre una coerenza equivalente al comportamento per strong-Global. Poiché questa coerenza richiede la disponibilità di più copie dei metadati degli oggetti in ogni sito, è possibile ricevere un elevato numero di errori del server interno 500 nel caso in cui due o più nodi storage nello stesso sito non fossero disponibili.

A meno che non si richiedano garanzie di coerenza simili a Amazon S3, è possibile evitare questi errori per le operazioni HEAD and GET impostando la coerenza su "disponibile". Quando un'operazione HEAD o GET utilizza la consistenza "disponibile", StorageGRID fornisce solo la consistenza finale. Non ritenta un'operazione non riuscita ad aumentare la coerenza, pertanto non richiede la disponibilità di più copie dei metadati degli oggetti.

specificare 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. Nell'esempio riportato di seguito viene impostata 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
```

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

specificare la coerenza per il bucket

Per impostare la coerenza per il bucket, è possibile utilizzare la richiesta StorageGRID"METTI la coerenza del bucket". In alternativa, è possibile "modificare la consistenza di un bucket" rivolgersi al responsabile del tenant.

Quando si imposta la consistenza per un secchio, tenere presente quanto segue:

- L'impostazione della consistenza per un bucket determina la consistenza utilizzata per S3 operazioni

eseguite sugli oggetti nel bucket o nella configurazione del bucket. Non influisce sulle operazioni sul bucket stesso.

- La coerenza per una singola operazione API sovrascrive la coerenza per il bucket.
- In generale, i bucket devono utilizzare la coerenza predefinita, "Read-after-new-write". Se le richieste non funzionano correttamente, modificare il comportamento del client dell'applicazione, se possibile. In alternativa, configurare il client per specificare la coerenza per ogni richiesta API. Impostare la consistenza a livello del bucket solo come ultima risorsa.

l'interazione tra coerenza e regole ILM per influire sulla protezione dei dati

Sia la scelta della coerenza che la regola ILM influiscono sulla protezione degli oggetti. Queste impostazioni possono interagire.

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

Per le regole ILM sono disponibili le seguenti "opzioni di acquisizione"opzioni:

Commit doppio
: StorageGRID effettua immediatamente copie provvisorie dell'oggetto e restituisce il successo al cliente. Le copie specificate nella regola ILM vengono eseguite quando possibile.

Rigoroso
: Tutte le copie specificate nella regola ILM devono essere eseguite prima che l'operazione sia restituita al cliente.

Bilanciato
: StorageGRID tenta di eseguire tutte le copie specificate nella regola ILM al momento dell'acquisizione; se ciò non è possibile, vengono create copie provvisorie e viene restituita al cliente l'avvenuta esecuzione. Le copie specificate nella regola ILM vengono eseguite quando possibile.

Esempio di interazione tra la regola coerenza e ILM

Si supponga di disporre di una griglia a due siti con la seguente regola ILM e la seguente coerenza:

- **ILM rule**: Creare due copie di oggetti, una nel sito locale e una in un sito remoto. USA un comportamento di acquisizione rigoroso.
- **Coerenza**: Strong-Global (i metadati degli oggetti vengono immediatamente distribuiti a tutti i siti).

Quando un client memorizza un oggetto nella griglia, StorageGRID esegue entrambe le copie degli oggetti e distribuisce i metadati a entrambi i siti prima di restituire il risultato al client.

L'oggetto è completamente protetto contro la perdita al momento dell'acquisizione del messaggio di successo. Ad esempio, se il sito locale viene perso poco dopo l'acquisizione, le copie dei dati dell'oggetto e dei metadati dell'oggetto rimangono nel sito remoto. L'oggetto è completamente recuperabile.

Se invece si è utilizzata la stessa regola ILM e la coerenza del sito sicuro, il client potrebbe ricevere un messaggio di successo dopo la replica dei dati dell'oggetto nel sito remoto ma prima della distribuzione dei metadati dell'oggetto. 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 poco dopo l'acquisizione, i metadati dell'oggetto andranno persi. Impossibile recuperare l'oggetto.

L'interrelazione tra coerenza e regole ILM può essere complessa. Contattare NetApp per assistenza.

Versione degli oggetti

È possibile impostare lo stato di versione di un bucket se si desidera mantenere più versioni di ciascun oggetto. L'abilitazione della versione per un bucket può aiutare a proteggere dalla cancellazione accidentale di oggetti e consente di recuperare e ripristinare le versioni precedenti di un oggetto.

Il sistema StorageGRID implementa il controllo delle versioni con il supporto per la maggior parte delle funzionalità e con alcune limitazioni. StorageGRID supporta fino a 10,000 versioni di ciascun oggetto.

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

L'utilizzo dell'autenticazione MFA (multi-factor Authentication) Delete non è supportato.

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

ILM e versione

I criteri ILM vengono applicati a ogni versione di un oggetto. Un processo di scansione ILM esegue una scansione continua di tutti gli oggetti e li rivaluti in base al criterio ILM corrente. Qualsiasi modifica apportata ai criteri ILM viene applicata a tutti gli oggetti precedentemente acquisiti. Sono incluse le versioni precedentemente ingerite se è abilitato il controllo delle versioni. La scansione ILM applica le nuove modifiche ILM agli oggetti acquisiti in precedenza.

Per gli oggetti S3 nei bucket abilitati per le versioni, il supporto per le versioni consente di creare regole ILM che utilizzano "tempo non corrente" come ora di riferimento (selezionare **Sì** per la domanda "Applica questa regola solo alle versioni precedenti degli oggetti?" in "Passaggio 1 della creazione guidata di una regola ILM"). Quando un oggetto viene aggiornato, le sue versioni precedenti diventano non aggiornate. L'utilizzo di un filtro "tempo non corrente" consente di creare policy per ridurre l'impatto sullo storage delle versioni precedenti di oggetti.

> Quando si carica una nuova versione di un oggetto utilizzando un'operazione di caricamento multiparte, l'ora non corrente per la versione originale dell'oggetto si riflette quando il caricamento multiparte è stato creato per la nuova versione, non quando il caricamento multiparte è stato completato. In casi limitati, il tempo non corrente per la versione originale potrebbe essere di ore o giorni prima del tempo per la versione corrente.

Informazioni correlate

- "Modalità di eliminazione degli oggetti con versione S3"
- "Regole e criteri ILM per gli oggetti con versione S3 (esempio 4)".

Utilizzare l'API REST S3 per configurare il blocco oggetti S3

Se l'impostazione blocco oggetti S3 globale è attivata per il sistema StorageGRID, è possibile creare bucket con blocco oggetti S3 attivato. È possibile specificare la conservazione predefinita per ogni bucket o impostazioni di conservazione per ciascuna versione dell'oggetto.

Come attivare il blocco oggetti S3 per un bucket

Se l'impostazione globale di blocco oggetti S3 è attivata per il sistema StorageGRID, è possibile attivare il blocco oggetti S3 quando si crea ciascun bucket.

S3 Object Lock è un'impostazione permanente che può essere attivata solo quando si crea un bucket. Non puoi aggiungere o disattivare il blocco oggetti S3 dopo la creazione di un bucket.

Per attivare il blocco oggetti S3 per un bucket, utilizzare uno dei seguenti metodi:

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

S3 Object Lock richiede il controllo della versione del bucket, che viene attivato automaticamente quando viene creato il bucket. Non puoi sospendere il controllo delle versioni per il bucket. Vedere "Versione degli oggetti".

Impostazioni di conservazione predefinite per un bucket

Quando S3 Object Lock è attivato per un bucket, è possibile attivare 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 a quando non viene raggiunta la data di conservazione.
  - La conservazione dell'oggetto fino alla data odierna 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 `s3:BypassGovernanceRetention` autorizzazione possono utilizzare l' `x-amz-bypass-governance-retention: true`intestazione della richiesta per ignorare le impostazioni di conservazione.
  - Questi utenti possono eliminare una versione dell'oggetto prima che venga raggiunta la data di conservazione.
  - Questi utenti possono aumentare, ridurre o rimuovere il mantenimento di un oggetto fino ad oggi.

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:

- Gestire le impostazioni del bucket da Tenant Manager. Vedere "Creare un bucket S3" e "Aggiorna la conservazione predefinita del blocco oggetti S3".
- Eseguire una richiesta PutObjectLockConfiguration per il 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 con blocco oggetti S3 attivato. È inoltre possibile rimuovere le impostazioni di conservazione predefinite precedentemente configurate.

Quando le nuove versioni degli oggetti vengono acquisite nel bucket, la modalità di conservazione predefinita viene applicata se `x-amz-object-lock-mode` e `x-amz-object-lock-retain-until-date` non sono specificate. Il periodo di conservazione predefinito viene utilizzato per calcolare la data di conservazione-fino-alla-data 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 della versione dell'oggetto rimane invariata e non viene ricalcolata utilizzando il nuovo periodo di conservazione predefinito.

Per completare questa operazione, è necessario disporre dell' `s3:PutBucketObjectLockConfiguration`autorizzazione o essere account root.

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

Esempio di richiesta

Questo esempio attiva il blocco oggetti S3 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 la conservazione predefinita per un bucket

Per determinare se S3 Object Lock è attivato per un bucket e per visualizzare la modalità di conservazione e il periodo di conservazione predefiniti, utilizzare uno dei seguenti metodi:

- Visualizza il bucket nel tenant manager. Vedere "Visualizza i bucket S3".
- Eseguire una richiesta GetObjectLockConfiguration.

GetObjectLockConfiguration

La richiesta GetObjectLockConfiguration consente di determinare se S3 Object Lock è attivato per un bucket e, se è attivato, verificare se sono presenti una modalità di conservazione predefinita e un periodo di conservazione configurato per il bucket.

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

Per completare questa operazione, è necessario disporre dell' `s3:GetBucketObjectLockConfiguration`autorizzazione o essere account root.

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 le impostazioni di conservazione S3 Object Lock.

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

È possibile specificare le seguenti impostazioni per ciascun oggetto:

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

    - In modalità COMPLIANCE, se la data di conservazione è futura, l'oggetto può essere recuperato, ma non può essere modificato o cancellato. È possibile aumentare la data di conservazione fino alla data prevista, ma non è possibile ridurla o rimuoverle.
    - In modalità GOVERNANCE, gli utenti con autorizzazioni speciali possono ignorare l'impostazione di conservazione fino alla data odierna. Possono eliminare una versione dell'oggetto prima che sia trascorso il periodo di conservazione. Possono anche aumentare, diminuire o addirittura rimuovere il mantenimento fino ad oggi.
- **Conservazione legale**: L'applicazione di un blocco legale a una versione oggetto blocca immediatamente tale oggetto. Ad esempio, potrebbe essere necessario sospendere legalmente un oggetto correlato a un'indagine o a una controversia legale. Una conservazione a fini giudiziari non ha una data di scadenza, ma rimane attiva fino a quando non viene esplicitamente rimossa.

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

Per specificare le impostazioni di blocco degli oggetti S3 quando si aggiunge una versione dell'oggetto a un bucket"PutObject", eseguire una richiesta , "Oggetto CopyObject"o "CreateMultipartUpload".

È possibile utilizzare quanto segue:

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

  > Se si specifica `x-amz-object-lock-mode`, è necessario 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 consentiti i secondi frazionari, ma vengono conservate solo 3 cifre decimali (precisione in millisecondi). Non sono consentiti altri formati ISO 8601.
    - La data di conservazione deve essere in futuro.
- `x-amz-object-lock-legal-hold`

  Se la conservazione legale È ATTIVA (sensibile al maiuscolo/minuscolo), l'oggetto viene collocato sotto una conservazione legale. Se l'opzione Legal Hold è disattivata, non viene effettuata alcuna conservazione a fini giudiziari. Qualsiasi altro valore genera un errore 400 Bad Request (InvalidArgument).

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

- L'`Content-MD5`intestazione della richiesta è necessaria se nella richiesta PutObject è presente un `x-amz-object-lock-*'intestazione di richiesta. `Content-MD5` Non è richiesto per CopyObject o CreateMultipartUpload.
- Se nel bucket non è abilitato il blocco oggetto S3 ed è presente un `x-amz-object-lock-*'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 abbinare il comportamento AWS. Tuttavia, quando un oggetto viene acquisito in un bucket con il blocco oggetti S3 attivato, StorageGRID eseguirà sempre un ingest a doppio commit.

- Una risposta successiva 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 configurate e se il mittente della richiesta dispone delle autorizzazioni corrette `s3:Get*`.

È possibile utilizzare il `s3:object-lock-remaining-retention-days` tasto delle condizioni della policy per limitare i periodi di conservazione minimo e massimo consentiti per gli oggetti.

Come aggiornare le impostazioni di conservazione per un oggetto

Se è necessario aggiornare le impostazioni di conservazione o conservazione a fini giudiziari per una versione di oggetto esistente, è possibile eseguire le seguenti operazioni di sottorisorsa oggetto:

- `PutObjectLegalHold`

  Se IL nuovo valore di conservazione a fini giudiziari è ATTIVO, l'oggetto viene collocato sotto una conservazione a fini giudiziari. Se il valore di conservazione a fini giudiziari è OFF, la conservazione a fini giudiziari viene revocata.

- `PutObjectRetention`
  - Il valore della modalità può essere COMPLIANCE o GOVERNANCE (distinzione tra maiuscole e minuscole).
  - Il valore Retain-until-date deve essere nel formato `2020-08-10T21:46:00Z`. Sono consentiti i secondi frazionari, ma vengono conservate solo 3 cifre decimali (precisione in millisecondi). Non sono consentiti altri formati ISO 8601.
  - Se una versione a oggetti ha un valore di conservazione esistente fino alla data odierna, è possibile aumentarlo. Il nuovo valore deve essere in futuro.

Come utilizzare LA modalità DI GOVERNANCE

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

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

  Non è possibile eliminare gli oggetti che si trovano sotto un blocco legale. La sospensione legale deve essere disattivata.

- Eseguire le operazioni PutObjectRetention che modificano la modalità di una versione dell'oggetto dalla GOVERNANCE alla CONFORMITÀ prima che sia trascorso il periodo di conservazione dell'oggetto.

  Non è mai consentito cambiare la modalità dalla CONFORMITÀ alla GOVERNANCE.

- Eseguire le operazioni PutObjectRetention per aumentare, ridurre o rimuovere il periodo di conservazione di una versione oggetto.

Informazioni correlate

- "Gestire gli oggetti con S3 Object Lock"
- "USA il blocco oggetti S3 per conservare gli oggetti"

- "Guida dell'utente di Amazon Simple Storage Service: Blocco degli oggetti"

Creare la configurazione del ciclo di vita S3

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

Il semplice esempio di questa sezione illustra come una configurazione del ciclo di vita S3 può controllare quando alcuni oggetti vengono cancellati (scaduti) da specifici bucket S3. L'esempio in questa sezione è a solo scopo illustrativo. Per informazioni dettagliate sulla creazione di configurazioni del ciclo di vita S3, vedere "Guida utente di Amazon Simple Storage Service: Gestione del ciclo di vita degli oggetti". Nota: StorageGRID supporta solo le azioni di scadenza e non 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 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: Consente di eliminare un oggetto quando viene raggiunta una data specificata o quando viene raggiunto un numero di giorni specificato, a partire dalla data di acquisizione dell'oggetto.
- NoncurrentVersionExpiration (NoncurrentExpiration versione): Consente di eliminare un oggetto quando viene raggiunto un numero di giorni specificato, a partire da quando l'oggetto è diventato non corrente.
- Filtro (prefisso, tag)
- Stato
- ID

Ciascun oggetto segue le impostazioni di conservazione di un ciclo di vita bucket S3 o di un criterio 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 corrispondenti 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 bucket e non sono specificate esplicitamente azioni di scadenza, le impostazioni di conservazione del criterio ILM non vengono utilizzate ed è implicito che le versioni degli oggetti vengano mantenute per sempre. Vedere "Esempi di priorità per il ciclo di vita dei bucket S3 e la politica ILM".

Di conseguenza, un oggetto potrebbe essere rimosso dalla griglia anche se le istruzioni di posizionamento in una regola ILM sono ancora applicabili all'oggetto. Oppure, un oggetto potrebbe essere conservato sulla griglia anche dopo che sono scadute le istruzioni di posizionamento ILM per l'oggetto. Per ulteriori informazioni, vedere "Come ILM opera per tutta la vita di un oggetto".

> La configurazione del ciclo di vita del bucket può essere utilizzata con bucket con blocco oggetti S3 attivato, ma la configurazione del ciclo di vita del bucket non è supportata per bucket conformi legacy.

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

- DeleteBucketLifecycle

- GetBucketLifecycleConfiguration
- PutBucketLifecycleConfiguration

Creare la configurazione del ciclo di vita

Come primo passo nella creazione di una configurazione del ciclo di vita, è possibile creare un file JSON che includa 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 corrispondenti 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 corrispondenti al filtro scadranno 100 giorni dopo l'acquisizione.

   > 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 che diventano 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, 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 verificare che una configurazione del ciclo di vita sia stata applicata correttamente al bucket, eseguire una richiesta GetBucketLifecycleConfiguration. Ad esempio:

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

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

Verificare 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 l'oggetto scade e quale regola di scadenza è stata abbinata.

> Poiché il ciclo di vita del bucket sovrascrive ILM, `expiry-date` viene visualizzata la data effettiva in cui l'oggetto verrà eliminato. Per ulteriori informazioni, vedere "Come viene determinata la conservazione degli oggetti".

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

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

La risposta corretta 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 corrisponde alla 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 per la versione, l' `x-amz-expiration`intestazione della risposta si applica solo alle versioni correnti di oggetti.

Raccomandazioni per l'implementazione dell'API REST S3

Seguire questi consigli quando si implementa l'API REST S3 per l'utilizzo con StorageGRID.

Raccomandazioni per la gestione di oggetti inesistenti

Se l'applicazione verifica regolarmente se un oggetto esiste in un percorso in cui non si prevede che l'oggetto esista effettivamente, è necessario utilizzare il comando "disponibile" "coerenza". Ad esempio, è necessario utilizzare la coerenza "disponibile" se l'applicazione rileva una posizione prima di INVIARLA.

In caso contrario, se l'operazione HEAD non trova l'oggetto, è possibile ricevere un numero elevato di errori del server interno 500 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 ciascun bucket utilizzando la richiesta oppure specificare la

coerenza nell'intestazione della "METTI la coerenza del bucket"richiesta per una singola operazione API.

Raccomandazioni per le chiavi a oggetti

Seguire questi consigli per i nomi delle chiavi degli oggetti, in base alla prima volta che il bucket è stato creato.

Bucket creati in StorageGRID 11.4 o versioni precedenti

- Non utilizzare valori casuali come primi quattro caratteri delle chiavi oggetto. Ciò è in contrasto con la precedente raccomandazione AWS per i prefissi principali. Utilizzare invece prefissi non casuali e non univoci, ad esempio `image` .
- Se si segue la precedente raccomandazione AWS per utilizzare caratteri casuali e univoci nei prefissi delle chiavi, inserire un prefisso tra le chiavi degli oggetti e il nome della directory. Ovvero, utilizzare 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 Best practice di performance. Nella maggior parte dei casi, è possibile utilizzare valori casuali per i primi quattro caratteri dei nomi delle chiavi degli oggetti.

> Un'eccezione 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 delle performance per questo caso d'utilizzo, modificare una parte iniziale del nome della chiave ogni diverse migliaia di oggetti con qualcosa di simile alla data. Si supponga, ad esempio, che un client S3 scriva in genere 2,000 oggetti al secondo e che il criterio del ciclo di vita di ILM o bucket rimuova tutti gli oggetti dopo tre giorni. Per ridurre al minimo l'impatto sulle prestazioni, è possibile assegnare un nome alle chiavi utilizzando uno schema simile al seguente: `/mybucket/mydir/yyyymmddhhmmss-random_UUID.jpg`

Raccomandazioni per "letture di gamma"

Se "opzione globale per comprimere gli oggetti memorizzati"è attivato, le applicazioni client S3 devono evitare di eseguire operazioni GetObject che specificano un intervallo di byte da restituire. Queste operazioni di "lettura dell'intervallo" sono inefficienti perché StorageGRID deve decomprimere efficacemente 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 da 50 GB.

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

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

Informazioni sul copyright

Informazioni sul marchio commerciale