So implementiert StorageGRID die S3 REST API

StorageGRID software

NetApp
October 21, 2025

Inhalt

So implementiert StorageGRID die S3 REST API

Widersprüchliche Clientanforderungen

Widersprüchliche Clientanforderungen, beispielsweise wenn zwei Clients auf denselben Schlüssel schreiben, werden nach dem Prinzip „Latest Wins“ gelöst.

Der Zeitpunkt für die Auswertung der „Latest Wins“ basiert darauf, wann das StorageGRID -System eine bestimmte Anfrage abschließt, und nicht darauf, wann S3-Clients einen Vorgang beginnen.

Konsistenzwerte

Konsistenz sorgt für ein Gleichgewicht zwischen der Verfügbarkeit der Objekte und der Konsistenz dieser Objekte über verschiedene Speicherknoten und Standorte hinweg. Sie können die Konsistenz je nach Anwendungsfall ändern.

Standardmäßig garantiert StorageGRID die Lese-nach-Schreib-Konsistenz für neu erstellte Objekte. Jeder GET nach einem erfolgreich abgeschlossenen PUT kann die neu geschriebenen Daten lesen. Überschreibungen vorhandener Objekte, Metadatenaktualisierungen und Löschungen sind letztendlich konsistent. Die Ausbreitung von Überschreibungen dauert im Allgemeinen Sekunden oder Minuten, kann aber bis zu 15 Tage dauern.

Wenn Sie Objektoperationen mit einer anderen Konsistenz durchführen möchten, können Sie:

- Geben Sie eine Konsistenz fürjeder Eimer .
- Geben Sie eine Konsistenz fürjede API-Operation .
- Ändern Sie die standardmäßige rasterweite Konsistenz, indem Sie eine der folgenden Aufgaben ausführen:
  - Gehen Sie im Grid Manager zu **KONFIGURATION** > **System** > **Speichereinstellungen** > **Standardkonsistenz**.
  - .

> Eine Änderung der rasterweiten Konsistenz gilt nur für Buckets, die nach der Änderung der Einstellung erstellt wurden. Um die Details einer Änderung zu ermitteln, sehen Sie sich das Audit-Protokoll an unter `/var/local/log` (Suche nach **Konsistenzebene**).

Konsistenzwerte

Die Konsistenz wirkt sich darauf aus, wie die Metadaten, die StorageGRID zum Verfolgen von Objekten verwendet, zwischen Knoten verteilt werden und somit auf die Verfügbarkeit von Objekten für Clientanforderungen.

Sie können die Konsistenz für einen Bucket oder eine API-Operation auf einen der folgenden Werte festlegen:

- **Alle**: Alle Knoten erhalten die Daten sofort, andernfalls schlägt die Anfrage fehl.
- **Stark global**: Garantiert Lese-nach-Schreib-Konsistenz für alle Clientanforderungen auf allen Sites.
- **Strong-Site**: Garantiert die Lese-nach-Schreib-Konsistenz für alle Clientanforderungen innerhalb einer

Site.

- **Lesen nach neuem Schreiben**: (Standard) Bietet Lese-nach-Schreib-Konsistenz für neue Objekte und letztendliche Konsistenz für Objektaktualisierungen. Bietet hohe Verfügbarkeit und Datenschutzgarantien. Für die meisten Fälle empfohlen.
- **Verfügbar**: Bietet letztendliche Konsistenz sowohl für neue Objekte als auch für Objektaktualisierungen. Verwenden Sie es für S3-Buckets nur nach Bedarf (z. B. für einen Bucket, der Protokollwerte enthält, die selten gelesen werden, oder für HEAD- oder GET-Operationen für nicht vorhandene Schlüssel). Wird für S3 FabricPool Buckets nicht unterstützt.

Verwenden Sie die Konsistenz „Lesen nach neuem Schreiben“ und „Verfügbar“.

Wenn ein HEAD- oder GET-Vorgang die Konsistenz „Lesen nach neuem Schreiben“ verwendet, führt StorageGRID die Suche in mehreren Schritten wie folgt durch:

- Es sucht zunächst mit geringer Konsistenz nach dem Objekt.
- Wenn diese Suche fehlschlägt, wird die Suche beim nächsten Konsistenzwert wiederholt, bis eine Konsistenz erreicht wird, die dem Verhalten für „stark global“ entspricht.

Wenn ein HEAD- oder GET-Vorgang die Konsistenz „Lesen nach neuem Schreiben“ verwendet, das Objekt jedoch nicht vorhanden ist, erreicht die Objektsuche immer eine Konsistenz, die dem Verhalten für „Strong-Global“ entspricht. Da für diese Konsistenz mehrere Kopien der Objektmetadaten an jedem Standort verfügbar sein müssen, kann es zu einer hohen Anzahl interner Serverfehler vom Typ 500 kommen, wenn zwei oder mehr Speicherknoten am selben Standort nicht verfügbar sind.

Sofern Sie keine Konsistenzgarantien ähnlich denen von Amazon S3 benötigen, können Sie diese Fehler bei HEAD- und GET-Vorgängen verhindern, indem Sie die Konsistenz auf „Verfügbar“ setzen. Wenn ein HEAD- oder GET-Vorgang die Konsistenz „Verfügbar“ verwendet, bietet StorageGRID nur die letztendliche Konsistenz. Ein fehlgeschlagener Vorgang wird bei zunehmender Konsistenz nicht wiederholt, sodass es nicht erforderlich ist, dass mehrere Kopien der Objektmetadaten verfügbar sind.

Konsistenz für API-Operation angeben

Um die Konsistenz für eine einzelne API-Operation festzulegen, müssen die Konsistenzwerte für die Operation unterstützt werden und Sie müssen die Konsistenz im Anforderungsheader angeben. In diesem Beispiel wird die Konsistenz für einen GetObject-Vorgang auf „Strong-Site“ festgelegt.

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

Sie müssen für die PutObject- und GetObject-Vorgänge dieselbe Konsistenz verwenden.

Konsistenz für Bucket angeben

Um die Konsistenz für den Bucket festzulegen, können Sie das StorageGRID verwenden"PUT Bucket-Konsistenz" Anfrage. Oder Sie können"die Konsistenz eines Eimers ändern" vom Mieterverwalter.

Beachten Sie beim Festlegen der Konsistenz für einen Bucket Folgendes:

- Durch das Festlegen der Konsistenz für einen Bucket wird bestimmt, welche Konsistenz für S3-Operationen verwendet wird, die an den Objekten im Bucket oder an der Bucket-Konfiguration ausgeführt werden. Es hat keine Auswirkungen auf Vorgänge am Bucket selbst.
- Die Konsistenz für einen einzelnen API-Vorgang überschreibt die Konsistenz für den Bucket.
- Im Allgemeinen sollten Buckets die Standardkonsistenz „Lesen nach neuem Schreiben“ verwenden. Wenn Anfragen nicht richtig funktionieren, ändern Sie nach Möglichkeit das Verhalten des Anwendungsclients. Oder konfigurieren Sie den Client so, dass die Konsistenz für jede API-Anforderung angegeben wird. Stellen Sie die Konsistenz auf Eimerebene nur als letztes Mittel ein.

[[Wie Konsistenzkontrollen und ILM-Regeln zusammenwirken]]Wie Konsistenz- und ILM-Regeln den Datenschutz beeinflussen

Sowohl Ihre Wahl der Konsistenz als auch Ihre ILM-Regel wirken sich darauf aus, wie Objekte geschützt werden. Diese Einstellungen können interagieren.

Beispielsweise wirkt sich die beim Speichern eines Objekts verwendete Konsistenz auf die anfängliche Platzierung der Objektmetadaten aus, während das für die ILM-Regel ausgewählte Aufnahmeverhalten die anfängliche Platzierung der Objektkopien beeinflusst. Da StorageGRID zur Erfüllung von Clientanforderungen Zugriff auf die Metadaten und Daten eines Objekts benötigt, kann die Auswahl passender Schutzebenen für Konsistenz und Aufnahmeverhalten einen besseren anfänglichen Datenschutz und vorhersehbarere Systemreaktionen bieten.

Die folgende"Aufnahmeoptionen" stehen für ILM-Regeln zur Verfügung:

Doppeltes Commit
: StorageGRID erstellt sofort Zwischenkopien des Objekts und meldet dem Client den Erfolg. Wenn möglich, werden die in der ILM-Regel angegebenen Kopien erstellt.

Strikt
: Alle in der ILM-Regel angegebenen Kopien müssen erstellt werden, bevor dem Client der Erfolg gemeldet wird.

Ausgewogen
: StorageGRID versucht, bei der Aufnahme alle in der ILM-Regel angegebenen Kopien zu erstellen. Wenn dies nicht möglich ist, werden Zwischenkopien erstellt und der Erfolg wird an den Client zurückgegeben. Die in der ILM-Regel angegebenen Kopien werden nach Möglichkeit erstellt.

Beispiel für die Interaktion zwischen Konsistenz- und ILM-Regel

Angenommen, Sie haben ein Grid mit zwei Sites mit der folgenden ILM-Regel und der folgenden Konsistenz:

- **ILM-Regel**: Erstellen Sie zwei Objektkopien, eine am lokalen Standort und eine an einem Remote-Standort. Verwenden Sie ein striktes Aufnahmeverhalten.
- **Konsistenz**: Stark global (Objektmetadaten werden sofort an alle Sites verteilt).

Wenn ein Client ein Objekt im Grid speichert, erstellt StorageGRID Kopien beider Objekte und verteilt Metadaten an beide Sites, bevor dem Client die Erfolgsmeldung zurückgegeben wird.

Zum Zeitpunkt der erfolgreichen Aufnahme der Nachricht ist das Objekt vollständig vor Verlust geschützt. Wenn beispielsweise die lokale Site kurz nach der Aufnahme verloren geht, sind am Remote-Standort weiterhin Kopien der Objektdaten und der Objektmetadaten vorhanden. Das Objekt ist vollständig abrufbar.

Wenn Sie stattdessen dieselbe ILM-Regel und die starke Site-Konsistenz verwenden, erhält der Client möglicherweise eine Erfolgsmeldung, nachdem die Objektdaten auf die Remote-Site repliziert wurden, aber bevor die Objektmetadaten dorthin verteilt werden. In diesem Fall entspricht das Schutzniveau der Objektmetadaten nicht dem Schutzniveau der Objektdaten. Wenn die lokale Site kurz nach der Aufnahme verloren geht, gehen die Objektmetadaten verloren. Das Objekt kann nicht abgerufen werden.

Die Wechselbeziehung zwischen Konsistenz und ILM-Regeln kann komplex sein. Wenden Sie sich an NetApp , wenn Sie Hilfe benötigen.

Objektversionierung

Sie können den Versionsstatus eines Buckets festlegen, wenn Sie mehrere Versionen jedes Objekts behalten möchten. Durch Aktivieren der Versionierung für einen Bucket können Sie vor dem versehentlichen Löschen von Objekten schützen und frühere Versionen eines Objekts abrufen und wiederherstellen.

Das StorageGRID -System implementiert Versionierung mit Unterstützung für die meisten Funktionen und mit einigen Einschränkungen. StorageGRID unterstützt bis zu 10.000 Versionen jedes Objekts.

Die Objektversionierung kann mit StorageGRID Information Lifecycle Management (ILM) oder mit der S3-Bucket-Lebenszykluskonfiguration kombiniert werden. Sie müssen die Versionierung für jeden Bucket explizit aktivieren. Wenn die Versionierung für einen Bucket aktiviert ist, wird jedem dem Bucket hinzugefügten Objekt eine Versions-ID zugewiesen, die vom StorageGRID -System generiert wird.

Die Verwendung von MFA (Multi-Faktor-Authentifizierung) zum Löschen wird nicht unterstützt.

> ⓘ Die Versionierung kann nur für Buckets aktiviert werden, die mit StorageGRID Version 10.3 oder höher erstellt wurden.

ILM und Versionierung

ILM-Richtlinien werden auf jede Version eines Objekts angewendet. Ein ILM-Scanprozess scannt kontinuierlich alle Objekte und wertet sie anhand der aktuellen ILM-Richtlinie neu aus. Alle Änderungen, die Sie an ILM-Richtlinien vornehmen, werden auf alle zuvor aufgenommenen Objekte angewendet. Dies schließt zuvor aufgenommene Versionen ein, wenn die Versionierung aktiviert ist. Beim ILM-Scannen werden neue ILM-Änderungen auf zuvor aufgenommene Objekte angewendet.

Für S3-Objekte in Buckets mit aktivierter Versionierung können Sie mit der Versionierungsunterstützung ILM-Regeln erstellen, die „Nicht aktuelle Zeit“ als Referenzzeit verwenden (wählen Sie **Ja** für die Frage „Diese Regel nur auf ältere Objektversionen anwenden?“ in"Schritt 1 des Assistenten „ILM-Regel erstellen““). Wenn ein Objekt aktualisiert wird, werden seine vorherigen Versionen nicht mehr aktuell. Mithilfe eines Filters „Nicht aktuelle Zeit“ können Sie Richtlinien erstellen, die die Speicherauswirkungen früherer Objektversionen reduzieren.

> ⓘ Wenn Sie eine neue Version eines Objekts mithilfe eines mehrteiligen Uploadvorgangs hochladen, gibt die nicht aktuelle Zeit für die ursprüngliche Version des Objekts an, wann der mehrteilige Upload für die neue Version erstellt wurde, und nicht, wann der mehrteilige Upload abgeschlossen wurde. In seltenen Fällen kann die nicht aktuelle Zeit der Originalversion Stunden oder Tage vor der Zeit der aktuellen Version liegen.

Ähnliche Informationen

- "So werden versionierte S3-Objekte gelöscht"
- "ILM-Regeln und -Richtlinien für versionierte S3-Objekte (Beispiel 4)" .

Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren

Wenn die globale S3-Objektsperre-Einstellung für Ihr StorageGRID System aktiviert ist, können Sie Buckets mit aktivierter S3-Objektsperre erstellen. Sie können die Standardaufbewahrung für jeden Bucket oder Aufbewahrungseinstellungen für jede Objektversion angeben.

So aktivieren Sie die S3-Objektsperre für einen Bucket

Wenn die globale S3-Objektsperre-Einstellung für Ihr StorageGRID System aktiviert ist, können Sie die S3-Objektsperre optional aktivieren, wenn Sie jeden Bucket erstellen.

S3 Object Lock ist eine permanente Einstellung, die nur aktiviert werden kann, wenn Sie einen Bucket erstellen. Sie können die S3-Objektsperre nicht hinzufügen oder deaktivieren, nachdem ein Bucket erstellt wurde.

Um die S3-Objektsperre für einen Bucket zu aktivieren, verwenden Sie eine der folgenden Methoden:

- Erstellen Sie den Bucket mit dem Tenant Manager. Sehen "S3-Bucket erstellen" .
- Erstellen Sie den Bucket mithilfe einer CreateBucket-Anforderung mit dem `x-amz-bucket-object-lock-enabled` Anforderungsheader. Sehen "Operationen an Buckets" .

S3 Object Lock erfordert eine Bucket-Versionierung, die beim Erstellen des Buckets automatisch aktiviert wird. Sie können die Versionsverwaltung für den Bucket nicht aussetzen. Sehen "Objektversionierung" .

Standardaufbewahrungseinstellungen für einen Bucket

Wenn S3 Object Lock für einen Bucket aktiviert ist, können Sie optional die Standardaufbewahrung für den Bucket aktivieren und einen Standardaufbewahrungsmodus und eine Standardaufbewahrungsdauer angeben.

Standardaufbewahrungsmodus

- Im COMPLIANCE-Modus:
  - Das Objekt kann erst gelöscht werden, wenn sein Aufbewahrungsdatum erreicht ist.
  - Das Aufbewahrungsdatum des Objekts kann erhöht, aber nicht verringert werden.
  - Das Aufbewahrungsdatum des Objekts kann erst entfernt werden, wenn dieses Datum erreicht ist.
- Im GOVERNANCE-Modus:
  - Benutzer mit der `s3:BypassGovernanceRetention` Berechtigung kann die `x-amz-bypass-governance-retention: true` Anforderungsheader zum Umgehen der Aufbewahrungseinstellungen.
  - Diese Benutzer können eine Objektversion löschen, bevor ihr Aufbewahrungsdatum erreicht ist.
  - Diese Benutzer können das Aufbewahrungsdatum eines Objekts erhöhen, verringern oder entfernen.

Standardaufbewahrungsdauer

Für jeden Bucket kann eine Standardaufbewahrungsdauer in Jahren oder Tagen angegeben werden.

So legen Sie die Standardaufbewahrung für einen Bucket fest

Um die Standardaufbewahrung für einen Bucket festzulegen, verwenden Sie eine der folgenden Methoden:

- Verwalten Sie die Bucket-Einstellungen über den Tenant Manager. Sehen"Erstellen eines S3-Buckets" Und"Standardaufbewahrung für S3 Object Lock aktualisieren" .
- Geben Sie eine PutObjectLockConfiguration-Anforderung für den Bucket aus, um den Standardmodus und die Standardanzahl von Tagen oder Jahren anzugeben.

PutObjectLockConfiguration

Mit der PutObjectLockConfiguration-Anforderung können Sie den Standardaufbewahrungsmodus und die Standardaufbewahrungsdauer für einen Bucket festlegen und ändern, bei dem S3 Object Lock aktiviert ist. Sie können auch zuvor konfigurierte Standardaufbewahrungseinstellungen entfernen.

Wenn neue Objektversionen in den Bucket aufgenommen werden, wird der Standardaufbewahrungsmodus angewendet, wenn `x-amz-object-lock-mode` Und `x-amz-object-lock-retain-until-date` sind nicht angegeben. Die Standardaufbewahrungsfrist wird zur Berechnung des Aufbewahrungs-bis-Datums verwendet, wenn `x-amz-object-lock-retain-until-date` ist nicht angegeben.

Wenn die Standardaufbewahrungsfrist nach der Aufnahme einer Objektversion geändert wird, bleibt das Aufbewahrungsdatum der Objektversion gleich und wird nicht anhand der neuen Standardaufbewahrungsfrist neu berechnet.

Sie müssen über die `s3:PutBucketObjectLockConfiguration` Berechtigung oder Root-Konto sein, um diesen Vorgang abzuschließen.

Der `Content-MD5` Der Anforderungsheader muss in der PUT-Anforderung angegeben werden.

Anforderungsbeispiel

Dieses Beispiel aktiviert S3 Object Lock für einen Bucket und legt den Standardaufbewahrungsmodus auf COMPLIANCE und die Standardaufbewahrungsdauer auf 6 Jahre fest.

```
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>
```

So bestimmen Sie die Standardaufbewahrung für einen Bucket

Um festzustellen, ob S3 Object Lock für einen Bucket aktiviert ist, und um den Standardaufbewahrungsmodus und die Aufbewahrungsdauer anzuzeigen, verwenden Sie eine der folgenden Methoden:

- Zeigen Sie den Bucket im Mandanten-Manager an. Sehen "S3-Buckets anzeigen" .
- Geben Sie eine GetObjectLockConfiguration-Anforderung aus.

GetObjectLockConfiguration

Mit der GetObjectLockConfiguration-Anforderung können Sie feststellen, ob die S3-Objektsperre für einen Bucket aktiviert ist. Wenn dies der Fall ist, können Sie prüfen, ob für den Bucket ein Standardaufbewahrungsmodus und eine Standardaufbewahrungsdauer konfiguriert sind.

Wenn neue Objektversionen in den Bucket aufgenommen werden, wird der Standardaufbewahrungsmodus angewendet, wenn `x-amz-object-lock-mode` ist nicht angegeben. Die Standardaufbewahrungsfrist wird zur Berechnung des Aufbewahrungs-bis-Datums verwendet, wenn `x-amz-object-lock-retain-until-date` ist nicht angegeben.

Sie müssen über die `s3:GetBucketObjectLockConfiguration` Berechtigung oder Root-Konto sein, um diesen Vorgang abzuschließen.

Anforderungsbeispiel

```
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
```

Antwortbeispiel

```
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>
```

So legen Sie Aufbewahrungseinstellungen für ein Objekt fest

Ein Bucket mit aktivierter S3 Object Lock kann eine Kombination aus Objekten mit und ohne S3 Object Lock-Aufbewahrungseinstellungen enthalten.

Aufbewahrungseinstellungen auf Objektebene werden mithilfe der S3 REST-API angegeben. Die Aufbewahrungseinstellungen für ein Objekt überschreiben alle Standardaufbewahrungseinstellungen für den Bucket.

Sie können für jedes Objekt die folgenden Einstellungen festlegen:

- **Aufbewahrungsmodus**: Entweder COMPLIANCE oder GOVERNANCE.
- **Aufbewahrungsdatum**: Ein Datum, das angibt, wie lange die Objektversion von StorageGRID aufbewahrt werden muss.

    - Wenn das Aufbewahrungsdatum im COMPLIANCE-Modus in der Zukunft liegt, kann das Objekt zwar abgerufen, aber nicht geändert oder gelöscht werden. Das Aufbewahrungsdatum kann verlängert werden, es kann jedoch nicht verkürzt oder entfernt werden.
    - Im GOVERNANCE-Modus können Benutzer mit Sonderberechtigung die Einstellung „Aufbewahren bis Datum“ umgehen. Sie können eine Objektversion löschen, bevor ihre Aufbewahrungsfrist abgelaufen ist. Sie können das Aufbewahrungsdatum auch verlängern, verkürzen oder sogar entfernen.
- **Rechtliche Sperre**: Durch Anwenden einer rechtlichen Sperre auf eine Objektversion wird dieses Objekt sofort gesperrt. Beispielsweise müssen Sie möglicherweise ein Objekt, das mit einer Untersuchung oder einem Rechtsstreit in Zusammenhang steht, rechtlich sperren. Eine rechtliche Sperre hat kein Ablaufdatum, sondern bleibt bestehen, bis sie ausdrücklich aufgehoben wird.

  Die Einstellung für die rechtliche Aufbewahrung eines Objekts ist unabhängig vom Aufbewahrungsmodus und dem Aufbewahrungsdatum. Wenn eine Objektversion einer rechtlichen Sperre unterliegt, kann niemand diese Version löschen.

Um S3 Object Lock-Einstellungen anzugeben, wenn Sie einem Bucket eine Objektversion hinzufügen, führen Sie einen"PutObject" ,"Objekt kopieren" , oder"CreateMultipartUpload" Anfrage.

Sie können Folgendes verwenden:

- `x-amz-object-lock-mode`, wobei COMPLIANCE oder GOVERNANCE (Groß-/Kleinschreibung beachten) lauten kann.

  > Wenn Sie angeben `x-amz-object-lock-mode` müssen Sie außerdem angeben `x-amz-object-lock-retain-until-date` .

- `x-amz-object-lock-retain-until-date`
    - Der Wert für das Retain-until-Datum muss das Format haben `2020-08-10T21:46:00Z` . Sekundenbruchteile sind zulässig, es bleiben jedoch nur 3 Dezimalstellen erhalten (Millisekundengenauigkeit). Andere ISO 8601-Formate sind nicht zulässig.
    - Das Aufbewahrungsdatum muss in der Zukunft liegen.
- `x-amz-object-lock-legal-hold`

  Wenn die rechtliche Sperre aktiviert ist (Groß-/Kleinschreibung beachten), wird das Objekt einer rechtlichen Sperre unterzogen. Wenn die rechtliche Sperre deaktiviert ist, wird keine rechtliche Sperre verhängt. Jeder andere Wert führt zu einem 400 Bad Request (InvalidArgument)-Fehler.

Wenn Sie einen dieser Anforderungsheader verwenden, beachten Sie die folgenden Einschränkungen:

- Der `Content-MD5` Anforderungsheader ist erforderlich, falls vorhanden `x-amz-object-lock-*` Der Anforderungsheader ist in der PutObject-Anforderung vorhanden. `Content-MD5` ist für CopyObject oder CreateMultipartUpload nicht erforderlich.
- Wenn für den Bucket die S3-Objektsperre nicht aktiviert ist und ein `x-amz-object-lock-*` Wenn kein Anforderungsheader vorhanden ist, wird der Fehler „400 Bad Request (InvalidRequest)“ zurückgegeben.
- Die PutObject-Anforderung unterstützt die Verwendung von `x-amz-storage-class: REDUCED_REDUNDANCY` um dem AWS-Verhalten zu entsprechen. Wenn jedoch ein Objekt in einen Bucket mit aktivierter S3-Objektsperre aufgenommen wird, führt StorageGRID immer eine Aufnahme mit doppeltem Commit durch.

- Eine nachfolgende GET- oder HeadObject-Versionsantwort enthält die Header `x-amz-object-lock-mode` , `x-amz-object-lock-retain-until-date` , Und `x-amz-object-lock-legal-hold` , sofern konfiguriert und der Absender der Anfrage über die richtige `s3:Get*` Berechtigungen.

Sie können die `s3:object-lock-remaining-retention-days` Richtlinienbedingungsschlüssel, um die minimal und maximal zulässigen Aufbewahrungsfristen für Ihre Objekte zu begrenzen.

So aktualisieren Sie die Aufbewahrungseinstellungen für ein Objekt

Wenn Sie die Einstellungen für die gesetzliche Aufbewahrungspflicht oder die Aufbewahrungsdauer für eine vorhandene Objektversion aktualisieren müssen, können Sie die folgenden Vorgänge für die Objektunterressource ausführen:

- `PutObjectLegalHold`

  Wenn der neue Wert für die rechtliche Sperre EIN ist, wird das Objekt einer rechtlichen Sperre unterzogen. Wenn der Legal-Hold-Wert auf „AUS“ gesetzt ist, wird der Legal Hold aufgehoben.

- `PutObjectRetention`
  - Der Moduswert kann COMPLIANCE oder GOVERNANCE sein (Groß-/Kleinschreibung beachten).
  - Der Wert für das Retain-until-Datum muss das Format haben `2020-08-10T21:46:00Z` . Sekundenbruchteile sind zulässig, es bleiben jedoch nur 3 Dezimalstellen erhalten (Millisekundengenauigkeit). Andere ISO 8601-Formate sind nicht zulässig.
  - Wenn für eine Objektversion ein vorhandenes Aufbewahrungsdatum vorhanden ist, können Sie dieses nur erhöhen. Der neue Wert muss in der Zukunft liegen.

So verwenden Sie den GOVERNANCE-Modus

Benutzer mit der `s3:BypassGovernanceRetention` Die Berechtigung kann die aktiven Aufbewahrungseinstellungen eines Objekts umgehen, das den GOVERNANCE-Modus verwendet. Alle DELETE- oder PutObjectRetention-Vorgänge müssen Folgendes enthalten: `x-amz-bypass-governance-retention:true` Anforderungsheader. Diese Benutzer können die folgenden zusätzlichen Vorgänge ausführen:

- Führen Sie die Vorgänge „DeleteObject“ oder „DeleteObjects“ aus, um eine Objektversion zu löschen, bevor ihre Aufbewahrungsfrist abgelaufen ist.

  Objekte, die einer rechtlichen Sperre unterliegen, können nicht gelöscht werden. Die rechtliche Sperre muss deaktiviert sein.

- Führen Sie PutObjectRetention-Vorgänge durch, die den Modus einer Objektversion von GOVERNANCE in COMPLIANCE ändern, bevor die Aufbewahrungsfrist des Objekts abgelaufen ist.

  Ein Wechsel des Modus von COMPLIANCE zu GOVERNANCE ist niemals zulässig.

- Führen Sie PutObjectRetention-Vorgänge durch, um die Aufbewahrungsdauer einer Objektversion zu erhöhen, zu verringern oder zu entfernen.

Ähnliche Informationen

- "Verwalten von Objekten mit S3 Object Lock"
- "Verwenden Sie S3 Object Lock, um Objekte beizubehalten"

- "Amazon Simple Storage Service-Benutzerhandbuch: Sperren von Objekten"

Erstellen einer S3-Lebenszykluskonfiguration

Sie können eine S3-Lebenszykluskonfiguration erstellen, um zu steuern, wann bestimmte Objekte aus dem StorageGRID System gelöscht werden.

Das einfache Beispiel in diesem Abschnitt veranschaulicht, wie eine S3-Lebenszykluskonfiguration steuern kann, wann bestimmte Objekte aus bestimmten S3-Buckets gelöscht werden (ablaufen). Das Beispiel in diesem Abschnitt dient nur zur Veranschaulichung. Ausführliche Informationen zum Erstellen von S3-Lebenszykluskonfigurationen finden Sie unter "Amazon Simple Storage Service-Benutzerhandbuch: Objekt-Lebenszyklusverwaltung" . Beachten Sie, dass StorageGRID nur Ablaufaktionen unterstützt, keine Übergangsaktionen.

Was ist eine Lebenszykluskonfiguration?

Eine Lebenszykluskonfiguration ist ein Satz von Regeln, die auf die Objekte in bestimmten S3-Buckets angewendet werden. Jede Regel gibt an, welche Objekte betroffen sind und wann diese Objekte ablaufen (an einem bestimmten Datum oder nach einer bestimmten Anzahl von Tagen).

StorageGRID unterstützt bis zu 1.000 Lebenszyklusregeln in einer Lebenszykluskonfiguration. Jede Regel kann die folgenden XML-Elemente enthalten:

- Ablauf: Löschen Sie ein Objekt, wenn ein bestimmtes Datum erreicht ist oder wenn eine bestimmte Anzahl von Tagen ab dem Zeitpunkt der Aufnahme des Objekts abgelaufen ist.
- NoncurrentVersionExpiration: Löschen Sie ein Objekt, wenn eine angegebene Anzahl von Tagen erreicht ist, beginnend mit dem Zeitpunkt, an dem das Objekt nicht mehr aktuell ist.
- Filter (Präfix, Tag)
- Status
- AUSWEIS

Jedes Objekt folgt den Aufbewahrungseinstellungen entweder eines S3-Bucket-Lebenszyklus oder einer ILM-Richtlinie. Wenn ein S3-Bucket-Lebenszyklus konfiguriert ist, überschreiben die Aktionen zum Ablauf des Lebenszyklus die ILM-Richtlinie für Objekte, die dem Bucket-Lebenszyklusfilter entsprechen. Objekte, die nicht dem Bucket-Lebenszyklusfilter entsprechen, verwenden die Aufbewahrungseinstellungen der ILM-Richtlinie. Wenn ein Objekt einem Bucket-Lebenszyklusfilter entspricht und keine Ablaufaktionen explizit angegeben sind, werden die Aufbewahrungseinstellungen der ILM-Richtlinie nicht verwendet und es wird davon ausgegangen, dass Objektversionen für immer aufbewahrt werden. Sehen "Beispielprioritäten für den S3-Bucket-Lebenszyklus und die ILM-Richtlinie" .

Dies kann dazu führen, dass ein Objekt aus dem Raster entfernt wird, obwohl die Platzierungsanweisungen in einer ILM-Regel weiterhin für das Objekt gelten. Oder ein Objekt kann auf dem Raster verbleiben, auch wenn alle ILM-Platzierungsanweisungen für das Objekt abgelaufen sind. Weitere Informationen finden Sie unter "Funktionsweise von ILM während der gesamten Lebensdauer eines Objekts" .

> Die Bucket-Lebenszykluskonfiguration kann mit Buckets verwendet werden, bei denen S3 Object Lock aktiviert ist. Für ältere konforme Buckets wird die Bucket-Lebenszykluskonfiguration jedoch nicht unterstützt.

StorageGRID unterstützt die Verwendung der folgenden Bucket-Operationen zur Verwaltung von Lebenszykluskonfigurationen:

- DeleteBucketLifecycle
- GetBucketLifecycleConfiguration
- PutBucketLifecycleConfiguration

Lebenszykluskonfiguration erstellen

Als ersten Schritt beim Erstellen einer Lebenszykluskonfiguration erstellen Sie eine JSON-Datei, die eine oder mehrere Regeln enthält. Diese JSON-Datei enthält beispielsweise die folgenden drei Regeln:

1. Regel 1 gilt nur für Objekte, die dem Präfix entsprechen `category1` / und die haben eine `key2` Wert von `tag2` . Der `Expiration` Der Parameter gibt an, dass Objekte, die dem Filter entsprechen, am 22. August 2020 um Mitternacht ablaufen.
2. Regel 2 gilt nur für Objekte, die dem Präfix entsprechen `category2` /. Der `Expiration` Der Parameter gibt an, dass Objekte, die dem Filter entsprechen, 100 Tage nach ihrer Aufnahme ablaufen.

   Regeln, die eine Anzahl von Tagen angeben, beziehen sich auf den Zeitpunkt der Aufnahme des Objekts. Wenn das aktuelle Datum das Aufnahmedatum plus die Anzahl der Tage überschreitet, werden einige Objekte möglicherweise aus dem Bucket entfernt, sobald die Lebenszykluskonfiguration angewendet wird.

3. Regel 3 gilt nur für Objekte, die dem Präfix entsprechen `category3` /. Der `Expiration` Der Parameter gibt an, dass alle nicht aktuellen Versionen übereinstimmender Objekte 50 Tage, nachdem sie nicht mehr aktuell sind, ablaufen.

```
{
    "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"
        }
    ]
}
```

Lebenszykluskonfiguration auf Bucket anwenden

Nachdem Sie die Lebenszyklus-Konfigurationsdatei erstellt haben, wenden Sie sie auf einen Bucket an, indem Sie eine PutBucketLifecycleConfiguration-Anforderung senden.

Diese Anfrage wendet die Lebenszykluskonfiguration in der Beispieldatei auf Objekte in einem Bucket namens `testbucket` .

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

Um zu überprüfen, ob eine Lebenszykluskonfiguration erfolgreich auf den Bucket angewendet wurde, senden Sie eine GetBucketLifecycleConfiguration-Anforderung. Beispiel:

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

Eine erfolgreiche Antwort listet die Lebenszykluskonfiguration auf, die Sie gerade angewendet haben.

Überprüfen Sie, ob das Ablaufdatum des Bucket-Lebenszyklus für das Objekt gilt

Sie können beim Ausgeben einer PutObject-, HeadObject- oder GetObject-Anforderung feststellen, ob eine Ablaufregel in der Lebenszykluskonfiguration auf ein bestimmtes Objekt zutrifft. Wenn eine Regel zutrifft, enthält die Antwort eine `Expiration` Parameter, der angibt, wann das Objekt abläuft und welche Ablaufregel erfüllt wurde.

> Da der Bucket-Lebenszyklus ILM außer Kraft setzt, `expiry-date` angezeigt wird das tatsächliche Datum, an dem das Objekt gelöscht wird. Weitere Informationen finden Sie unter "So wird die Objektaufbewahrung bestimmt" .

Beispielsweise wurde diese PutObject-Anforderung am 22. Juni 2020 ausgegeben und platziert ein Objekt in der `testbucket` Eimer.

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

Die Erfolgsantwort gibt an, dass das Objekt in 100 Tagen (01. Oktober 2020) abläuft und dass es Regel 2 der Lebenszykluskonfiguration entspricht.

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

Beispielsweise wurde diese HeadObject-Anforderung verwendet, um Metadaten für dasselbe Objekt im Testbucket-Bucket abzurufen.

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

Die Erfolgsantwort enthält die Metadaten des Objekts und gibt an, dass das Objekt in 100 Tagen abläuft und Regel 2 entspricht.

```
{
      "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": {}
}
```

Für Buckets mit aktivierter Versionierung gilt: `x-amz-expiration` Der Antwortheader gilt nur für aktuelle Versionen von Objekten.

Empfehlungen zur Implementierung der S3 REST API

Sie sollten diese Empfehlungen befolgen, wenn Sie die S3 REST-API zur Verwendung mit StorageGRID implementieren.

Empfehlungen für HEADs zu nicht vorhandenen Objekten

Wenn Ihre Anwendung routinemäßig prüft, ob ein Objekt an einem Pfad existiert, an dem Sie das Objekt nicht erwarten, sollten Sie die Option "Verfügbar" verwenden."Konsistenz" . Sie sollten beispielsweise die Konsistenz „Verfügbar“ verwenden, wenn Ihre Anwendung einen HEAD für einen Speicherort vor dem PUT anwendet.

Andernfalls kann es vorkommen, dass Sie, wenn der HEAD-Vorgang das Objekt nicht findet, eine große Anzahl interner Serverfehler vom Typ 500 erhalten, wenn zwei oder mehr Speicherknoten am selben Standort nicht verfügbar sind oder ein Remote-Standort nicht erreichbar ist.

Sie können die "Verfügbare" Konsistenz für jeden Bucket mithilfe der"PUT Bucket-Konsistenz" Anfrage, oder Sie können die Konsistenz im Anfrageheader für eine einzelne API-Operation angeben.

Empfehlungen für Objektschlüssel

Befolgen Sie diese Empfehlungen für Objektschlüsselnamen, basierend auf dem Zeitpunkt der ersten Erstellung des Buckets.

Buckets, die in StorageGRID 11.4 oder früher erstellt wurden

- Verwenden Sie keine zufälligen Werte als die ersten vier Zeichen der Objektschlüssel. Dies steht im Gegensatz zur früheren AWS-Empfehlung für Schlüsselpräfixe. Verwenden Sie stattdessen nicht zufällige, nicht eindeutige Präfixe, wie etwa `image` .
- Wenn Sie der früheren AWS-Empfehlung folgen, zufällige und eindeutige Zeichen in Schlüsselpräfixen zu verwenden, stellen Sie den Objektschlüsseln einen Verzeichnisnamen voran. Das heißt, verwenden Sie dieses Format:

  `mybucket/mydir/f8e3-image3132.jpg`

  Anstelle dieses Formats:

  `mybucket/f8e3-image3132.jpg`

In StorageGRID 11.4 oder höher erstellte Buckets

Eine Einschränkung der Objektschlüsselnamen zur Einhaltung der Best Practices für die Leistung ist nicht erforderlich. In den meisten Fällen können Sie für die ersten vier Zeichen von Objektschlüsselnamen zufällige Werte verwenden.

> Eine Ausnahme hiervon stellt ein S3-Workload dar, der kontinuierlich alle Objekte nach kurzer Zeit entfernt. Um die Auswirkungen auf die Leistung in diesem Anwendungsfall zu minimieren, variieren Sie alle paar tausend Objekte einen führenden Teil des Schlüsselnamens mit etwas wie dem Datum. Nehmen wir beispielsweise an, dass ein S3-Client normalerweise 2.000 Objekte/Sekunde schreibt und die ILM- oder Bucket-Lebenszyklusrichtlinie alle Objekte nach drei Tagen entfernt. Um die Auswirkungen auf die Leistung zu minimieren, können Sie Schlüssel nach einem Muster wie diesem benennen: `/mybucket/mydir/`*`yyyymmddhhmmss-random_UUID`*`.jpg`

Empfehlungen für „Range Reads“

Wenn die"globale Option zum Komprimieren gespeicherter Objekte" aktiviert ist, sollten S3-Clientanwendungen die Durchführung von GetObject-Operationen vermeiden, die einen zurückzugebenden Bytebereich angeben. Diese „Range Read“-Operationen sind ineffizient, da StorageGRID die Objekte effektiv dekomprimieren muss, um auf die angeforderten Bytes zuzugreifen. GetObject-Operationen, die einen kleinen Bytebereich aus einem sehr großen Objekt anfordern, sind besonders ineffizient. Beispielsweise ist es ineffizient, einen 10 MB großen Bereich aus einem komprimierten 50 GB-Objekt zu lesen.

Wenn Bereiche aus komprimierten Objekten gelesen werden, kann es bei Clientanforderungen zu einer Zeitüberschreitung kommen.

> Wenn Sie Objekte komprimieren müssen und Ihre Clientanwendung Bereichslesevorgänge verwenden muss, erhöhen Sie das Lesezeitlimit für die Anwendung.