Verwenden Sie die S3 REST-API

StorageGRID software

NetApp
October 21, 2025

Inhalt

Verwenden Sie die S3 REST-API

Unterstützte Versionen und Updates der S3 REST API

StorageGRID unterstützt die Simple Storage Service (S3)-API, die als Satz von Representational State Transfer (REST)-Webdiensten implementiert ist.

Durch die Unterstützung der S3 REST API können Sie serviceorientierte Anwendungen, die für S3-Webdienste entwickelt wurden, mit lokalem Objektspeicher verbinden, der das StorageGRID -System verwendet. Es sind nur minimale Änderungen an der aktuellen Verwendung von S3 REST-API-Aufrufen durch eine Clientanwendung erforderlich.

Unterstützte Versionen

StorageGRID unterstützt die folgenden spezifischen Versionen von S3 und HTTP.

Artikel	Version
S3-API-Spezifikation	"Amazon Web Services (AWS)-Dokumentation: Amazon Simple Storage Service API-Referenz"
HTTP	1,1 Weitere Informationen zu HTTP finden Sie unter HTTP/1.1 (RFCs 7230-35). "IETF RFC 2616: Hypertext Transfer Protocol (HTTP/1.1)" **Hinweis**: StorageGRID unterstützt kein HTTP/1.1-Pipelining.

Aktualisierungen der S3 REST API-Unterstützung

Freigeben	Kommentare
11,9	• Unterstützung für vorberechnete SHA-256-Prüfsummenwerte für die folgenden Anfragen und unterstützten Header hinzugefügt. Mit dieser Funktion können Sie die Integrität hochgeladener Objekte überprüfen: ◦ CompleteMultipartUpload: `x-amz-checksum-sha256` ◦ CreateMultipartUpload: `x-amz-checksum-algorithm` ◦ GetObject: `x-amz-checksum-mode` ◦ Kopfobjekt: `x-amz-checksum-mode` ◦ Teileliste ◦ PutObject: `x-amz-checksum-sha256` ◦ UploadPart: `x-amz-checksum-sha256` • Dem Grid-Administrator wurde die Möglichkeit hinzugefügt, die Aufbewahrungs- und Compliance-Einstellungen auf Mandantenebene zu steuern. Diese Einstellungen wirken sich auf die S3 Object Lock-Einstellungen aus. ◦ Standardaufbewahrungsmodus für Buckets und Objektaufbewahrungsmodus: Governance oder Compliance, sofern vom Grid-Administrator zugelassen. ◦ Standardaufbewahrungszeitraum des Buckets und Aufbewahrungsdatum des Objekts: Muss kleiner oder gleich dem zulässigen maximalen Aufbewahrungszeitraum sein, der vom Grid-Administrator festgelegt wurde. • Verbesserte Unterstützung für `aws-chunked` Inhaltskodierung und Streaming `x-amz-content-sha256` Werte. Einschränkungen: ◦ Falls vorhanden, `chunk-signature` ist optional und nicht validiert ◦ Falls vorhanden, `x-amz-trailer` Inhalt wird ignoriert
11,8	Die Namen der S3-Operationen wurden aktualisiert, damit sie mit den Namen übereinstimmen, die in der "Amazon Web Services (AWS)-Dokumentation: Amazon Simple Storage Service API-Referenz" .
11,7	• Hinzugefügt"Kurzreferenz: Unterstützte S3-API-Anfragen" . • Unterstützung für die Verwendung des GOVERNANCE-Modus mit S3 Object Lock hinzugefügt. • Unterstützung für das StorageGRID-spezifische `x-ntap-sg-cgr-replication-status` Antwortheader für GET Object- und HEAD Object-Anfragen. Dieser Header gibt den Replikationsstatus eines Objekts für die Cross-Grid-Replikation an. • SelectObjectContent-Anfragen unterstützen jetzt Parquet-Objekte.

Freigeben	Kommentare
11,6	• Unterstützung für die Verwendung von hinzugefügt `partNumber` Anforderungsparameter in GET-Objekt- und HEAD-Objektanforderungen. • Unterstützung für einen Standardaufbewahrungsmodus und eine Standardaufbewahrungsdauer auf Bucket-Ebene für S3 Object Lock hinzugefügt. • Zusätzliche Unterstützung für die `s3:object-lock-remaining-retention-days` Richtlinienbedingungsschlüssel, um den Bereich der zulässigen Aufbewahrungszeiträume für Ihre Objekte festzulegen. • Die maximal *empfohlene* Größe für einen einzelnen PUT-Objektvorgang wurde auf 5 GiB (5.368.709.120 Bytes) geändert. Wenn Sie Objekte haben, die größer als 5 GiB sind, verwenden Sie stattdessen den mehrteiligen Upload.
11,5	• Unterstützung für die Verwaltung der Bucket-Verschlüsselung hinzugefügt. • Unterstützung für S3 Object Lock und veraltete Compliance-Anfragen hinzugefügt. • Unterstützung für die Verwendung von DELETE Multiple Objects bei versionierten Buckets hinzugefügt. • Der `Content-MD5` Der Anforderungsheader wird jetzt korrekt unterstützt.
11,4	• Unterstützung für DELETE-Bucket-Tagging, GET-Bucket-Tagging und PUT-Bucket-Tagging hinzugefügt. Kostenzuordnungs-Tags werden nicht unterstützt. • Für in StorageGRID 11.4 erstellte Buckets ist die Einschränkung von Objektschlüsselnamen zur Einhaltung der Best Practices für die Leistung nicht mehr erforderlich. • Unterstützung für Bucket-Benachrichtigungen hinzugefügt auf der `s3:ObjectRestore:Post` Ereignistyp. • AWS-Größenbeschränkungen für mehrteilige Teile werden jetzt erzwungen. Jeder Teil eines mehrteiligen Uploads muss zwischen 5 MiB und 5 GiB groß sein. Der letzte Teil kann kleiner als 5 MiB sein. • Unterstützung für TLS 1.3 hinzugefügt
11,3	• Unterstützung für die serverseitige Verschlüsselung von Objektdaten mit vom Kunden bereitgestellten Schlüsseln (SSE-C) hinzugefügt. • Unterstützung für DELETE-, GET- und PUT-Bucket-Lebenszyklusoperationen (nur Ablaufaktion) und für die `x-amz-expiration` Antwortheader. • PUT-Objekt, PUT-Objekt – Kopieren und mehrteiliger Upload aktualisiert, um die Auswirkungen von ILM-Regeln zu beschreiben, die eine synchrone Platzierung bei der Aufnahme verwenden. • TLS 1.1-Chiffren werden nicht mehr unterstützt.

Freigeben	Kommentare
11,2	Unterstützung für die POST-Objektwiederherstellung zur Verwendung mit Cloud-Speicherpools hinzugefügt. Unterstützung für die Verwendung der AWS-Syntax für ARN, Richtlinienbedingungsschlüssel und Richtlinienvariablen in Gruppen- und Bucket-Richtlinien hinzugefügt. Vorhandene Gruppen- und Bucket-Richtlinien, die die StorageGRID Syntax verwenden, werden weiterhin unterstützt. **Hinweis:** Die Verwendung von ARN/URN in anderen JSON/XML-Konfigurationen, einschließlich der in benutzerdefinierten StorageGRID -Funktionen verwendeten, hat sich nicht geändert.
11,1	Unterstützung für Cross-Origin Resource Sharing (CORS), HTTP für S3-Clientverbindungen zu Grid-Knoten und Compliance-Einstellungen für Buckets hinzugefügt.
11,0	Unterstützung für die Konfiguration von Plattformdiensten (CloudMirror-Replikation, Benachrichtigungen und Elasticsearch-Suchintegration) für Buckets hinzugefügt. Außerdem wurde Unterstützung für die Objektmarkierung, Standortbeschränkungen für Buckets und die verfügbare Konsistenz hinzugefügt.
10,4	Unterstützung für ILM-Scan-Änderungen an der Versionierung, Aktualisierungen der Seite „Endpoint Domain Names“, Bedingungen und Variablen in Richtlinien, Richtlinienbeispielen und der Berechtigung „PutOverwriteObject“ hinzugefügt.
10,3	Unterstützung für Versionierung hinzugefügt.
10,2	Unterstützung für Gruppen- und Bucket-Zugriffsrichtlinien sowie für mehrteilige Kopien (Upload-Teil – Kopie) hinzugefügt.
10,1	Unterstützung für mehrteilige Uploads, Anfragen im virtuellen gehosteten Stil und v4-Authentifizierung hinzugefügt.
10,0	Erste Unterstützung der S3 REST API durch das StorageGRID -System. Die aktuell unterstützte Version der *Simple Storage Service API Reference* ist der 01.03.2006.

Kurzreferenz: Unterstützte S3-API-Anfragen

Auf dieser Seite wird zusammengefasst, wie StorageGRID die APIs des Amazon Simple Storage Service (S3) unterstützt.

Diese Seite enthält nur die S3-Operationen, die von StorageGRID unterstützt werden.

Um die AWS-Dokumentation für jeden Vorgang anzuzeigen, wählen Sie den Link in der Überschrift aus.

Allgemeine URI-Abfrageparameter und Anforderungsheader

Sofern nicht anders angegeben, werden die folgenden allgemeinen URI-Abfrageparameter unterstützt:

- `versionId`(wie für Objektoperationen erforderlich)

Sofern nicht anders angegeben, werden die folgenden allgemeinen Anforderungsheader unterstützt:

- `Authorization`
- `Connection`
- `Content-Length`
- `Content-MD5`
- `Content-Type`
- `Date`
- `Expect`
- `Host`
- `x-amz-date`

Ähnliche Informationen

- "Details zur S3 REST API-Implementierung"
- "Amazon Simple Storage Service API-Referenz: Allgemeine Anforderungsheader"

"AbortMultipartUpload"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage, plus diesen zusätzlichen URI-Abfrageparameter:

- `uploadId`

Anforderungstext

Keine

StorageGRID -Dokumentation

"Vorgänge für mehrteilige Uploads"

"CompleteMultipartUpload"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage, plus diesen zusätzlichen URI-Abfrageparameter:

- `uploadId`
- `x-amz-checksum-sha256`

XML-Tags des Anforderungstexts

StorageGRID unterstützt die folgenden XML-Tags im Anforderungstext:

- `ChecksumSHA256`
- `CompleteMultipartUpload`
- `ETag`
- `Part`
- `PartNumber`

StorageGRID -Dokumentation

"CompleteMultipartUpload"

"Objekt kopieren"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage, plus diese zusätzlichen Header:

- `x-amz-copy-source`
- `x-amz-copy-source-if-match`
- `x-amz-copy-source-if-modified-since`
- `x-amz-copy-source-if-none-match`
- `x-amz-copy-source-if-unmodified-since`
- `x-amz-copy-source-server-side-encryption-customer-algorithm`
- `x-amz-copy-source-server-side-encryption-customer-key`
- `x-amz-copy-source-server-side-encryption-customer-key-MD5`
- `x-amz-metadata-directive`
- `x-amz-object-lock-legal-hold`
- `x-amz-object-lock-mode`
- `x-amz-object-lock-retain-until-date`
- `x-amz-server-side-encryption`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `x-amz-storage-class`
- `x-amz-tagging`
- `x-amz-tagging-directive`
- `x-amz-meta-<metadata-name>`

Anforderungstext

Keine

StorageGRID -Dokumentation

"Objekt kopieren"

"Bucket erstellen"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allegemeine Parameter und Header für diese Anfrage, plus diese zusätzlichen Header:

- `x-amz-bucket-object-lock-enabled`

Anforderungstext

StorageGRID unterstützt alle Anforderungstextparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST-API definiert wurden.

StorageGRID -Dokumentation

"Operationen an Buckets"

"CreateMultipartUpload"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt alleallgemeine Parameter und Header für diese Anfrage, plus diese zusätzlichen Header:

- `Cache-Control`
- `Content-Disposition`
- `Content-Encoding`
- `Content-Language`
- `Expires`
- `x-amz-checksum-algorithm`
- `x-amz-server-side-encryption`
- `x-amz-storage-class`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `x-amz-tagging`
- `x-amz-object-lock-mode`
- `x-amz-object-lock-retain-until-date`
- `x-amz-object-lock-legal-hold`
- `x-amz-meta-<metadata-name>`

Anforderungstext

Keine

StorageGRID -Dokumentation

"CreateMultipartUpload"

"Bucket löschen"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

StorageGRID -Dokumentation

"Operationen an Buckets"

"BucketCors löschen"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"DeleteBucketEncryption"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"DeleteBucketLifecycle"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

- "Operationen an Buckets"
- "Erstellen einer S3-Lebenszykluskonfiguration"

"DeleteBucketPolicy"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"DeleteBucketReplication"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"BucketTagging löschen"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"Objekt löschen"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage, plus diesen zusätzlichen Anfrageheader:

- `x-amz-bypass-governance-retention`

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Objekten"

"Objekte löschen"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage, plus diesen zusätzlichen Anfrageheader:

- `x-amz-bypass-governance-retention`

Anforderungstext

StorageGRID unterstützt alle Anforderungstextparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST-API definiert wurden.

StorageGRID -Dokumentation

"Operationen an Objekten"

"DeleteObjectTagging"

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Objekten"

"GetBucketAcl"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"GetBucketCors"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"GetBucketEncryption"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"GetBucketLifecycleConfiguration"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

- "Operationen an Buckets"
- "Erstellen einer S3-Lebenszykluskonfiguration"

"BucketLocation abrufen"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"GetBucketNotificationConfiguration"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"GetBucketPolicy"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"GetBucketReplication"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allegemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"GetBucketTagging"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allegemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"GetBucketVersioning"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allegemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"GetObject"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allegemeine Parameter und Header für diese Anfrage, plus diese zusätzlichen URI-Abfrageparameter:

- `x-amz-checksum-mode`
- `partNumber`
- `response-cache-control`
- `response-content-disposition`

- `response-content-encoding`
- `response-content-language`
- `response-content-type`
- `response-expires`

Und diese zusätzlichen Anforderungsheader:

- `Range`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `If-Match`
- `If-Modified-Since`
- `If-None-Match`
- `If-Unmodified-Since`

Anforderungstext

Keine

StorageGRID -Dokumentation

"GetObject"

"GetObjectAcl"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Objekten"

"GetObjectLegalHold"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

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

"GetObjectLockConfiguration"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allegemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

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

"GetObjectRetention"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allegemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

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

"GetObjectTagging"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allegemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Objekten"

"Kopfeimer"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allegemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"HeadObject"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allegemeine Parameter und Header für diese Anfrage, plus diese zusätzlichen Header:

- `x-amz-checksum-mode`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `If-Match`
- `If-Modified-Since`
- `If-None-Match`
- `If-Unmodified-Since`
- `Range`

Anforderungstext

Keine

StorageGRID -Dokumentation

"HeadObject"

"Buckets auflisten"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allegemeine Parameter und Header für diese Anfrage.

Anforderungstext

Keine

StorageGRID -Dokumentation

Operationen auf dem Dienst › ListBuckets

"ListMultipartUploads"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allegemeine Parameter und Header für diese Anfrage, plus diese zusätzlichen Parameter:

- `encoding-type`
- `key-marker`
- `max-uploads`
- `prefix`
- `upload-id-marker`

Anforderungstext

Keine

StorageGRID -Dokumentation

"ListMultipartUploads"

"ListObjects"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allealigemeine Parameter und Header für diese Anfrage, plus diese zusätzlichen Parameter:

- `delimiter`
- `encoding-type`
- `marker`
- `max-keys`
- `prefix`

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"ListObjectsV2"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt alleallgemeine Parameter und Header für diese Anfrage, plus diese zusätzlichen Parameter:

- `continuation-token`
- `delimiter`
- `encoding-type`
- `fetch-owner`
- `max-keys`
- `prefix`
- `start-after`

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"ListObjectVersions"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt alleallgemeine Parameter und Header für diese Anfrage, plus diese zusätzlichen Parameter:

- `delimiter`

- `encoding-type`
- `key-marker`
- `max-keys`
- `prefix`
- `version-id-marker`

Anforderungstext

Keine

StorageGRID -Dokumentation

"Operationen an Buckets"

"Teileliste"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage, plus diese zusätzlichen Parameter:

- `max-parts`
- `part-number-marker`
- `uploadId`

Anforderungstext

Keine

StorageGRID -Dokumentation

"ListMultipartUploads"

"PutBucketCors"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

StorageGRID unterstützt alle Anforderungstextparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST-API definiert wurden.

StorageGRID -Dokumentation

"Operationen an Buckets"

"PutBucketEncryption"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

XML-Tags des Anforderungstexts

StorageGRID unterstützt die folgenden XML-Tags im Anforderungstext:

- `ApplyServerSideEncryptionByDefault`
- `Rule`
- `ServerSideEncryptionConfiguration`
- `SSEAlgorithm`

StorageGRID -Dokumentation

"Operationen an Buckets"

"PutBucketLifecycleConfiguration"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

XML-Tags des Anforderungstexts

StorageGRID unterstützt die folgenden XML-Tags im Anforderungstext:

- `And`
- `Days`
- `Expiration`
- `ExpiredObjectDeleteMarker`
- `Filter`
- `ID`
- `Key`
- `LifecycleConfiguration`
- `NewerNoncurrentVersions`
- `NoncurrentDays`
- `NoncurrentVersionExpiration`
- `Prefix`
- `Rule`
- `Status`
- `Tag`
- `Value`

StorageGRID -Dokumentation

- "Operationen an Buckets"
- "Erstellen einer S3-Lebenszykluskonfiguration"

"PutBucketNotificationConfiguration"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

XML-Tags des Anforderungstexts

StorageGRID unterstützt die folgenden XML-Tags im Anforderungstext:

- `Event`
- `Filter`
- `FilterRule`
- `Id`
- `Name`
- `NotificationConfiguration`
- `Prefix`
- `S3Key`
- `Suffix`
- `Topic`
- `TopicConfiguration`
- `Value`

StorageGRID -Dokumentation

"Operationen an Buckets"

"PutBucketPolicy"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Einzelheiten zu den unterstützten JSON-Body-Feldern finden Sie unter"Verwenden Sie Bucket- und Gruppenzugriffsrichtlinien" .

"PutBucketReplication"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

XML-Tags des Anforderungstexts

- `Bucket`
- `Destination`
- `Prefix`
- `ReplicationConfiguration`

- `Rule`
- `Status`
- `StorageClass`

StorageGRID -Dokumentation

"Operationen an Buckets"

"PutBucketTagging"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstext

StorageGRID unterstützt alle Anforderungstextparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST-API definiert wurden.

StorageGRID -Dokumentation

"Operationen an Buckets"

"PutBucketVersioning"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage.

Anforderungstextparameter

StorageGRID unterstützt die folgenden Anforderungstextparameter:

- `VersioningConfiguration`
- `Status`

StorageGRID -Dokumentation

"Operationen an Buckets"

"PutObject"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage, plus diese zusätzlichen Header:

- `Cache-Control`
- `Content-Disposition`
- `Content-Encoding`
- `Content-Language`
- `Expires`
- `x-amz-checksum-sha256`
- `x-amz-server-side-encryption`

- `x-amz-storage-class`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `x-amz-tagging`
- `x-amz-object-lock-mode`
- `x-amz-object-lock-retain-until-date`
- `x-amz-object-lock-legal-hold`
- `x-amz-meta-<metadata-name>`

Anforderungstext

- Binärdaten des Objekts

StorageGRID -Dokumentation

"PutObject"

"PutObjectLegalHold"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt alleallgemeine Parameter und Header für diese Anfrage.

Anforderungstext

StorageGRID unterstützt alle Anforderungstextparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST-API definiert wurden.

StorageGRID -Dokumentation

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

"PutObjectLockConfiguration"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt alleallgemeine Parameter und Header für diese Anfrage.

Anforderungstext

StorageGRID unterstützt alle Anforderungstextparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST-API definiert wurden.

StorageGRID -Dokumentation

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

"PutObjectRetention"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt alleallgemeine Parameter und Header für diese Anfrage, plus diesen zusätzlichen Header:

- `x-amz-bypass-governance-retention`

Anforderungstext

StorageGRID unterstützt alle Anforderungstextparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST-API definiert wurden.

StorageGRID -Dokumentation

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

"PutObjectTagging"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allealgemeine Parameter und Header für diese Anfrage.

Anforderungstext

StorageGRID unterstützt alle Anforderungstextparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST-API definiert wurden.

StorageGRID -Dokumentation

"Operationen an Objekten"

"RestoreObject"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt alleallgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Einzelheiten zu den unterstützten Body-Feldern finden Sie unter"RestoreObject" .

"SelectObjectContent"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt alleallgemeine Parameter und Header für diese Anfrage.

Anforderungstext

Einzelheiten zu den unterstützten Textfeldern finden Sie hier:

- "Verwenden Sie S3 Select"
- "SelectObjectContent"

"UploadPart"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt alleallgemeine Parameter und Header für diese Anfrage, plus diese zusätzlichen URI-Abfrageparameter:

- `partNumber`
- `uploadId`

Und diese zusätzlichen Anforderungsheader:

- `x-amz-checksum-sha256`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`

Anforderungstext

- Binärdaten des Teils

StorageGRID -Dokumentation

"UploadPart"

"UploadPartCopy"

URI-Abfrageparameter und Anforderungsheader

StorageGRID unterstützt allgemeine Parameter und Header für diese Anfrage, plus diese zusätzlichen URI-Abfrageparameter:

- `partNumber`
- `uploadId`

Und diese zusätzlichen Anforderungsheader:

- `x-amz-copy-source`
- `x-amz-copy-source-if-match`
- `x-amz-copy-source-if-modified-since`
- `x-amz-copy-source-if-none-match`
- `x-amz-copy-source-if-unmodified-since`
- `x-amz-copy-source-range`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`
- `x-amz-copy-source-server-side-encryption-customer-algorithm`
- `x-amz-copy-source-server-side-encryption-customer-key`
- `x-amz-copy-source-server-side-encryption-customer-key-MD5`

Anforderungstext

Keine

StorageGRID -Dokumentation

"UploadPartCopy"

Testen der S3 REST API-Konfiguration

Sie können die Amazon Web Services Command Line Interface (AWS CLI) verwenden, um Ihre Verbindung zum System zu testen und zu überprüfen, ob Sie Objekte lesen und schreiben können.

Bevor Sie beginnen

- Sie haben die AWS CLI von heruntergeladen und installiert "aws.amazon.com/cli" .
- Optional haben Sie"einen Load Balancer-Endpunkt erstellt" . Andernfalls kennen Sie die IP-Adresse des Speicherknotens, mit dem Sie eine Verbindung herstellen möchten, und die zu verwendende Portnummer. Sehen "IP-Adressen und Ports für Clientverbindungen" .
- Du hast"ein S3-Mandantenkonto erstellt" .
- Sie haben sich beim Mandanten angemeldet und"einen Zugriffsschlüssel erstellt" .

Einzelheiten zu diesen Schritten finden Sie unter"Konfigurieren von Clientverbindungen" .

Schritte

1. Konfigurieren Sie die AWS CLI-Einstellungen, um das Konto zu verwenden, das Sie im StorageGRID -System erstellt haben:
   a. Wechseln Sie in den Konfigurationsmodus: `aws configure`
   b. Geben Sie die Zugriffsschlüssel-ID für das von Ihnen erstellte Konto ein.
   c. Geben Sie den geheimen Zugriffsschlüssel für das von Ihnen erstellte Konto ein.
   d. Geben Sie die zu verwendende Standardregion ein. Beispiel: `us-east-1` .
   e. Geben Sie das zu verwendende Standardausgabeformat ein oder drücken Sie die Eingabetaste, um JSON auszuwählen.
2. Erstellen Sie einen Bucket.

   In diesem Beispiel wird davon ausgegangen, dass Sie einen Load Balancer-Endpunkt für die Verwendung der IP-Adresse 10.96.101.17 und des Ports 10443 konfiguriert haben.

   ```
   aws s3api --endpoint-url https://10.96.101.17:10443
   --no-verify-ssl create-bucket --bucket testbucket
   ```

   Wenn der Bucket erfolgreich erstellt wurde, wird der Speicherort des Buckets zurückgegeben, wie im folgenden Beispiel zu sehen ist:

   ```
   "Location": "/testbucket"
   ```

3. Laden Sie ein Objekt hoch.

   ```
   aws s3api --endpoint-url https://10.96.101.17:10443 --no-verify-ssl
   put-object --bucket testbucket --key s3.pdf --body C:\s3-
   test\upload\s3.pdf
   ```

Wenn das Objekt erfolgreich hochgeladen wurde, wird ein Etag zurückgegeben, der ein Hash der Objektdaten ist.

4. Listen Sie den Inhalt des Buckets auf, um zu überprüfen, ob das Objekt hochgeladen wurde.

```
aws s3api --endpoint-url https://10.96.101.17:10443 --no-verify-ssl
list-objects --bucket testbucket
```

5. Löschen Sie das Objekt.

```
aws s3api --endpoint-url https://10.96.101.17:10443 --no-verify-ssl
delete-object --bucket testbucket --key s3.pdf
```

6. Löschen Sie den Bucket.

```
aws s3api --endpoint-url https://10.96.101.17:10443 --no-verify-ssl
delete-bucket --bucket testbucket
```

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.

Unterstützung für Amazon S3 REST API

Details zur S3 REST API-Implementierung

Das StorageGRID -System implementiert die Simple Storage Service API (API-Version 2006-03-01) mit Unterstützung für die meisten Vorgänge und mit einigen Einschränkungen. Sie müssen die Implementierungsdetails verstehen, wenn Sie S3 REST API-Clientanwendungen integrieren.

Das StorageGRID -System unterstützt sowohl Anfragen im virtuellen gehosteten Stil als auch Anfragen im Pfadstil.

Datumsverarbeitung

Die StorageGRID -Implementierung der S3 REST API unterstützt nur gültige HTTP-Datumsformate.

Das StorageGRID -System unterstützt nur gültige HTTP-Datumsformate für Header, die Datumswerte akzeptieren. Der Zeitanteil des Datums kann im Format Greenwich Mean Time (GMT) oder im Format Universal Coordinated Time (UTC) ohne Zeitzonenverschiebung angegeben werden (+0000 muss angegeben werden). Wenn Sie die `x-amz-date` Header in Ihrer Anfrage, überschreibt es alle im Date-Anforderungsheader angegebenen Werte. Bei Verwendung von AWS Signature Version 4 ist die `x-amz-date` Der Header muss in der signierten Anfrage vorhanden sein, da der Datumsheader nicht unterstützt wird.

Allgemeine Anforderungsheader

Das StorageGRID -System unterstützt die gemeinsamen Anforderungsheader, die definiert sind durch "Amazon Simple Storage Service API-Referenz: Allgemeine Anforderungsheader" , mit einer Ausnahme.

Anforderungsheader	Durchführung
Genehmigung	Volle Unterstützung für AWS Signature Version 2 Unterstützung für AWS Signature Version 4, mit folgenden Ausnahmen: • Wenn Sie den tatsächlichen Nutzlastprüfsummenwert in `x-amz-content-sha256` wird der Wert ohne Validierung akzeptiert, als ob der Wert `UNSIGNED-PAYLOAD` für den Header vorgesehen war. Wenn Sie eine `x-amz-content-sha256` Header-Wert, der impliziert `aws-chunked` Streaming (z. B. STREAMING-AWS4-HMAC-SHA256-PAYLOAD), werden die Chunk-Signaturen nicht anhand der Chunk-Daten überprüft.
x-amz-Sicherheitstoken	Nicht implementiert. Rückgaben `XNotImplemented` .

Allgemeine Antwortheader

Das StorageGRID -System unterstützt alle gängigen Antwortheader, die in der *Simple Storage Service API Reference* definiert sind, mit einer Ausnahme.

Antwortheader	Durchführung
x-amz-id-2	Nicht verwendet

Authentifizieren von Anfragen

Das StorageGRID -System unterstützt sowohl authentifizierten als auch anonymen Zugriff auf Objekte mithilfe der S3-API.

Die S3-API unterstützt Signature Version 2 und Signature Version 4 zur Authentifizierung von S3-API-Anfragen.

Authentifizierte Anfragen müssen mit Ihrer Zugriffsschlüssel-ID und Ihrem geheimen Zugriffsschlüssel signiert werden.

Das StorageGRID -System unterstützt zwei Authentifizierungsmethoden: HTTP `Authorization` Header und Verwendung von Abfrageparametern.

Verwenden Sie den HTTP-Autorisierungsheader

Das HTTP `Authorization` Der Header wird von allen S3-API-Operationen verwendet, außer von anonymen Anfragen, sofern dies durch die Bucket-Richtlinie zulässig ist. Der `Authorization` Der Header enthält alle erforderlichen Signaturinformationen zur Authentifizierung einer Anfrage.

Verwenden von Abfrageparametern

Sie können Abfrageparameter verwenden, um einer URL Authentifizierungsinformationen hinzuzufügen. Dies wird als Vorsignierung der URL bezeichnet und kann verwendet werden, um vorübergehenden Zugriff auf bestimmte Ressourcen zu gewähren. Benutzer mit der vorsignierten URL müssen den geheimen Zugriffsschlüssel nicht kennen, um auf die Ressource zuzugreifen. Dadurch können Sie Dritten eingeschränkten Zugriff auf eine Ressource gewähren.

Vorgänge für den Dienst

Das StorageGRID -System unterstützt die folgenden Vorgänge für den Dienst.

Betrieb	Durchführung
Buckets auflisten (früher GET-Dienst genannt)	Implementiert mit dem gesamten Amazon S3 REST API-Verhalten. Änderungen vorbehalten.
GET-Speichernutzung	Das StorageGRID"GET-Speichernutzung" Die Anfrage gibt Auskunft über die Gesamtmenge des von einem Konto und jedem mit dem Konto verknüpften Bucket verwendeten Speichers. Dies ist eine Operation für den Dienst mit einem Pfad von / und einem benutzerdefinierten Abfrageparameter(`?x-ntap-sg-usage`) hinzugefügt.

Betrieb	Durchführung
OPTIONEN /	Clientanwendungen können `OPTIONS /` Anfragen an den S3-Port eines Speicherknotens, ohne S3-Authentifizierungsdaten anzugeben, um festzustellen, ob der Speicherknoten verfügbar ist. Sie können diese Anfrage zur Überwachung verwenden oder um externen Lastenausgleichsmodulen zu ermöglichen, zu erkennen, wenn ein Speicherknoten ausgefallen ist.

Operationen an Buckets

Das StorageGRID -System unterstützt maximal 5.000 Buckets für jedes S3-Mandantenkonto.

Jedes Raster kann maximal 100.000 Buckets enthalten.

Um 5.000 Buckets zu unterstützen, muss jeder Speicherknoten im Grid über mindestens 64 GB RAM verfügen.

Die Einschränkungen für Bucket-Namen folgen den regionalen Einschränkungen des AWS-US-Standards, Sie sollten sie jedoch zusätzlich auf DNS-Namenskonventionen beschränken, um Anfragen im virtuellen S3-Hosting-Stil zu unterstützen.

Weitere Informationen finden Sie im Folgenden:

- "Amazon Simple Storage Service-Benutzerhandbuch: Bucket-Kontingente, Einschränkungen und Begrenzungen"
- "Konfigurieren von S3-Endpunktdomänennamen"

Die Operationen ListObjects (GET Bucket) und ListObjectVersions (GET Bucket-Objektversionen) unterstützen StorageGRID"Konsistenzwerte" .

Sie können überprüfen, ob Aktualisierungen der letzten Zugriffszeit für einzelne Buckets aktiviert oder deaktiviert sind. Sehen "GET Bucket – Letzte Zugriffszeit" .

Die folgende Tabelle beschreibt, wie StorageGRID S3 REST API-Bucket-Operationen implementiert. Um diese Vorgänge auszuführen, müssen die erforderlichen Zugangsdaten für das Konto angegeben werden.

Betrieb	Durchführung
Bucket erstellen	Erstellt einen neuen Bucket. Indem Sie den Bucket erstellen, werden Sie zum Bucket-Eigentümer. • Bucket-Namen müssen den folgenden Regeln entsprechen: ◦ Muss in jedem StorageGRID -System eindeutig sein (nicht nur innerhalb des Mandantenkontos). ◦ Muss DNS-kompatibel sein. ◦ Muss mindestens 3 und darf nicht mehr als 63 Zeichen enthalten. ◦ Kann eine Reihe von einem oder mehreren Labels sein, wobei benachbarte Labels durch einen Punkt getrennt sind. Jedes Etikett muss mit einem Kleinbuchstaben oder einer Zahl beginnen und enden und darf nur Kleinbuchstaben, Zahlen und Bindestriche enthalten. ◦ Darf nicht wie eine IP-Adresse im Textformat aussehen. ◦ In Anfragen im virtuell gehosteten Stil sollten keine Punkte verwendet werden. Punkte verursachen Probleme bei der Überprüfung des Platzhalterzertifikats des Servers. • Standardmäßig werden Buckets im `us-east-1` Region; Sie können jedoch die `LocationConstraint` Anforderungselement im Anforderungstext, um eine andere Region anzugeben. Bei Verwendung der `LocationConstraint` Element müssen Sie den genauen Namen einer Region angeben, die mit dem Grid Manager oder der Grid Management API definiert wurde. Wenden Sie sich an Ihren Systemadministrator, wenn Sie den zu verwendenden Regionsnamen nicht kennen. **Hinweis**: Es tritt ein Fehler auf, wenn Ihre CreateBucket-Anforderung eine Region verwendet, die nicht in StorageGRID definiert wurde. • Sie können Folgendes einschließen: `x-amz-bucket-object-lock-enabled` Anforderungsheader zum Erstellen eines Buckets mit aktivierter S3-Objektsperre. Sehen "Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren" . Sie müssen S3 Object Lock aktivieren, wenn Sie den Bucket erstellen. Sie können die S3-Objektsperre nicht hinzufügen oder deaktivieren, nachdem ein Bucket erstellt wurde. S3 Object Lock erfordert eine Bucket-Versionierung, die automatisch aktiviert wird, wenn Sie den Bucket erstellen.
Bucket löschen	Löscht den Bucket.
BucketCors löschen	Löscht die CORS-Konfiguration für den Bucket.
DeleteBucketEncryption	Löscht die Standardverschlüsselung aus dem Bucket. Vorhandene verschlüsselte Objekte bleiben verschlüsselt, aber alle neuen Objekte, die dem Bucket hinzugefügt werden, werden nicht verschlüsselt.

Betrieb	Durchführung
DeleteBucketLifecycle	Löscht die Lebenszykluskonfiguration aus dem Bucket. Sehen "Erstellen einer S3-Lebenszykluskonfiguration" .
DeleteBucketPolicy	Löscht die an den Bucket angehängte Richtlinie.
DeleteBucketReplication	Löscht die an den Bucket angehängte Replikationskonfiguration.
BucketTagging löschen	Verwendet die `tagging` Unterressource zum Entfernen aller Tags aus einem Bucket. **Achtung**: Wenn für diesen Bucket ein nicht standardmäßiger ILM-Richtlinientag festgelegt ist, wird ein `NTAP-SG-ILM-BUCKET-TAG` Bucket-Tag mit einem ihm zugewiesenen Wert. Geben Sie keine DeleteBucketTagging-Anforderung aus, wenn ein `NTAP-SG-ILM-BUCKET-TAG` Eimer-Tag. Senden Sie stattdessen eine PutBucketTagging-Anfrage mit nur dem `NTAP-SG-ILM-BUCKET-TAG` Tag und sein zugewiesener Wert, um alle anderen Tags aus dem Bucket zu entfernen. Verändern oder entfernen Sie nicht die `NTAP-SG-ILM-BUCKET-TAG` Eimer-Tag.
GetBucketAcl	Gibt eine positive Antwort sowie die ID, den Anzeigenamen und die Berechtigung des Bucket-Eigentümers zurück und gibt damit an, dass der Eigentümer vollen Zugriff auf den Bucket hat.
GetBucketCors	Gibt den `cors` Konfiguration für den Bucket.
GetBucketEncryption	Gibt die Standardverschlüsselungskonfiguration für den Bucket zurück.
GetBucketLifecycleConfiguration (früher GET Bucket-Lebenszyklus genannt)	Gibt die Lebenszykluskonfiguration für den Bucket zurück. Sehen "Erstellen einer S3-Lebenszykluskonfiguration" .
BucketLocation abrufen	Gibt die Region zurück, die mit dem `LocationConstraint` Element in der CreateBucket-Anforderung. Wenn die Region des Buckets `us-east-1` , wird für die Region eine leere Zeichenfolge zurückgegeben.
GetBucketNotificationConfiguration (früher „GET Bucket-Benachrichtigung“ genannt)	Gibt die dem Bucket zugeordnete Benachrichtigungskonfiguration zurück.
GetBucketPolicy	Gibt die dem Bucket zugeordnete Richtlinie zurück.
GetBucketReplication	Gibt die dem Bucket zugeordnete Replikationskonfiguration zurück.

Betrieb	Durchführung
GetBucketTagging	Verwendet die `tagging` Unterressource, um alle Tags für einen Bucket zurückzugeben. **Achtung**: Wenn für diesen Bucket ein nicht standardmäßiger ILM-Richtlinientag festgelegt ist, wird ein `NTAP-SG-ILM-BUCKET-TAG` Bucket-Tag mit einem ihm zugewiesenen Wert. Ändern oder entfernen Sie dieses Tag nicht.
GetBucketVersioning	Diese Implementierung verwendet die `versioning` Unterressource, um den Versionsstatus eines Buckets zurückzugeben. • *blank*: Die Versionierung wurde nie aktiviert (Bucket ist „Unversioned“) • Aktiviert: Versionierung ist aktiviert • Ausgesetzt: Die Versionsverwaltung war zuvor aktiviert und ist ausgesetzt
GetObjectLockConfiguration	Gibt den Standardaufbewahrungsmodus und die Standardaufbewahrungsdauer des Buckets zurück, sofern konfiguriert. Sehen "Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren" .
Kopfeimer	Stellt fest, ob ein Bucket vorhanden ist und Sie über die Berechtigung verfügen, darauf zuzugreifen. Dieser Vorgang gibt Folgendes zurück: • `x-ntap-sg-bucket-id`: Die UUID des Buckets im UUID-Format. • `x-ntap-sg-trace-id`: Die eindeutige Trace-ID der zugehörigen Anfrage.
ListObjects und ListObjectsV2 (früher GET Bucket genannt)	Gibt einige oder alle (bis zu 1.000) Objekte in einem Bucket zurück. Die Speicherklasse für Objekte kann einen von zwei Werten haben, auch wenn das Objekt mit dem `REDUCED_REDUNDANCY` Speicherklassenoption: • `STANDARD`, was darauf hinweist, dass das Objekt in einem Speicherpool gespeichert ist, der aus Speicherknoten besteht. • `GLACIER`, was darauf hinweist, dass das Objekt in den vom Cloud Storage Pool angegebenen externen Bucket verschoben wurde. Wenn der Bucket eine große Anzahl gelöschter Schlüssel mit demselben Präfix enthält, kann die Antwort einige `CommonPrefixes` die keine Schlüssel enthalten.
ListObjectVersions (zuvor GET Bucket Object-Versionen genannt)	Mit Lesezugriff auf einen Bucket kann dieser Vorgang mit dem `versions` Die Unterressource listet Metadaten aller Versionen von Objekten im Bucket auf.

Betrieb	Durchführung
PutBucketCors	Legt die CORS-Konfiguration für einen Bucket fest, sodass der Bucket Cross-Origin-Anfragen verarbeiten kann. Cross-Origin Resource Sharing (CORS) ist ein Sicherheitsmechanismus, der es Client-Webanwendungen in einer Domäne ermöglicht, auf Ressourcen in einer anderen Domäne zuzugreifen. Angenommen, Sie verwenden einen S3-Bucket namens `images` zum Speichern von Grafiken. Durch Festlegen der CORS-Konfiguration für die `images` Bucket, können Sie die Anzeige der Bilder in diesem Bucket auf der Website zulassen `http://www.example.com` .
PutBucketEncryption	Legt den Standardverschlüsselungsstatus eines vorhandenen Buckets fest. Wenn die Verschlüsselung auf Bucket-Ebene aktiviert ist, werden alle neuen Objekte, die dem Bucket hinzugefügt werden, verschlüsselt. StorageGRID unterstützt serverseitige Verschlüsselung mit von StorageGRID verwalteten Schlüsseln. Wenn Sie die serverseitige Verschlüsselungskonfigurationsregel angeben, legen Sie die `SSEAlgorithm` Parameter auf `AES256` und verwenden Sie nicht die `KMSMasterKeyID` Parameter. Die Standardverschlüsselungskonfiguration des Buckets wird ignoriert, wenn die Objekt-Upload-Anforderung bereits eine Verschlüsselung angibt (d. h. wenn die Anforderung die `x-amz-server-side-encryption-*` Anforderungsheader).
PutBucketLifecycleConfiguration (früher PUT Bucket-Lebenszyklus genannt)	Erstellt eine neue Lebenszykluskonfiguration für den Bucket oder ersetzt eine vorhandene Lebenszykluskonfiguration. StorageGRID unterstützt bis zu 1.000 Lebenszyklusregeln in einer Lebenszykluskonfiguration. Jede Regel kann die folgenden XML-Elemente enthalten: • Ablauf (Tage, Datum, ExpiredObjectDeleteMarker) • NoncurrentVersionExpiration (NewerNoncurrentVersions, NoncurrentDays) • Filter (Präfix, Tag) • Status • AUSWEIS StorageGRID unterstützt diese Aktionen nicht: • AbbruchUnvollständigMehrteiliger Upload • Übergang Sehen "Erstellen einer S3-Lebenszykluskonfiguration" . Informationen dazu, wie die Ablaufaktion in einem Bucket-Lebenszyklus mit ILM-Platzierungsanweisungen interagiert, finden Sie unter"Funktionsweise von ILM während der gesamten Lebensdauer eines Objekts" . **Hinweis**: Die Bucket-Lebenszykluskonfiguration kann mit Buckets verwendet werden, bei denen S3 Object Lock aktiviert ist, die Bucket-Lebenszykluskonfiguration wird jedoch für ältere konforme Buckets nicht unterstützt.

Betrieb	Durchführung

PutBucketNotificationConfiguration
 	
(früher PUT Bucket-Benachrichtigung genannt)
	Konfiguriert Benachrichtigungen für den Bucket mithilfe der im Anforderungstext enthaltenen Benachrichtigungskonfigurations-XML. Sie sollten sich der folgenden Implementierungsdetails bewusst sein:
	
	• StorageGRID unterstützt Amazon Simple Notification Service (Amazon SNS) oder Kafka-Themen als Ziele. Simple Queue Service (SQS) oder Amazon Lambda-Endpunkte werden nicht unterstützt.
	• Das Ziel für Benachrichtigungen muss als URN eines StorageGRID Endpunkts angegeben werden. Endpunkte können mit dem Tenant Manager oder der Tenant Management API erstellt werden.
	
	Der Endpunkt muss vorhanden sein, damit die Benachrichtigungskonfiguration erfolgreich ist. Wenn der Endpunkt nicht existiert, wird ein `400 Bad Request` Fehler wird mit dem Code zurückgegeben `InvalidArgument` .
	
	• Für die folgenden Ereignistypen können Sie keine Benachrichtigung konfigurieren. Diese Ereignistypen werden **nicht** unterstützt.
	◦ `s3:ReducedRedundancyLostObject`
	◦ `s3:ObjectRestore:Completed`
	• Von StorageGRID gesendete Ereignisbenachrichtigungen verwenden das standardmäßige JSON-Format, mit der Ausnahme, dass sie einige Schlüssel nicht enthalten und für andere bestimmte Werte verwenden, wie in der folgenden Liste gezeigt:
	◦ **Ereignisquelle**
	`sgws:s3`
	◦ **awsRegion**
	nicht enthalten
	◦ **x-amz-id-2**
	nicht enthalten
	◦ **arn**
	`urn:sgws:s3:::bucket_name`

PutBucketPolicy	Legt die dem Bucket zugeordnete Richtlinie fest. Sehen "Verwenden Sie Bucket- und Gruppenzugriffsrichtlinien" .

Betrieb	Durchführung
PutBucketReplication	Konfiguriert"StorageGRID CloudMirror-Replikation" für den Bucket unter Verwendung der im Anforderungstext bereitgestellten XML-Replikationskonfiguration. Bei der CloudMirror-Replikation sollten Sie die folgenden Implementierungsdetails beachten: • StorageGRID unterstützt nur V1 der Replikationskonfiguration. Dies bedeutet, dass StorageGRID die Verwendung des `Filter` Element für Regeln und befolgt V1-Konventionen zum Löschen von Objektversionen. Weitere Einzelheiten finden Sie unter "Amazon Simple Storage Service-Benutzerhandbuch: Replikationskonfiguration" . • Die Bucket-Replikation kann für versionierte oder nicht versionierte Buckets konfiguriert werden. • Sie können in jeder Regel der XML-Replikationskonfiguration einen anderen Ziel-Bucket angeben. Ein Quell-Bucket kann in mehr als einen Ziel-Bucket repliziert werden. • Ziel-Buckets müssen als URN von StorageGRID -Endpunkten angegeben werden, wie im Tenant Manager oder der Tenant Management API angegeben. Sehen "Konfigurieren der CloudMirror-Replikation" . Der Endpunkt muss vorhanden sein, damit die Replikationskonfiguration erfolgreich ist. Wenn der Endpunkt nicht existiert, schlägt die Anfrage fehl, da `400 Bad Request` Die Fehlermeldung lautet: `Unable to save the replication policy. The specified endpoint URN does not exist: URN.` • Sie müssen kein `Role` im Konfigurations-XML. Dieser Wert wird von StorageGRID nicht verwendet und wird ignoriert, wenn er übermittelt wird. • Wenn Sie die Speicherklasse aus der Konfigurations-XML weglassen, verwendet StorageGRID die `STANDARD` Speicherklasse standardmäßig. • Wenn Sie ein Objekt aus dem Quell-Bucket oder den Quell-Bucket selbst löschen, ist das regionsübergreifende Replikationsverhalten wie folgt: ◦ Wenn Sie das Objekt oder den Bucket löschen, bevor es repliziert wurde, wird das Objekt/der Bucket nicht repliziert und Sie werden nicht benachrichtigt. ◦ Wenn Sie das Objekt oder den Bucket nach der Replikation löschen, folgt StorageGRID dem standardmäßigen Löschverhalten von Amazon S3 für V1 der regionsübergreifenden Replikation.

Betrieb	Durchführung
PutBucketTagging	Verwendet die `tagging` Unterressource zum Hinzufügen oder Aktualisieren eines Satzes von Tags für einen Bucket. Beachten Sie beim Hinzufügen von Bucket-Tags die folgenden Einschränkungen: • Sowohl StorageGRID als auch Amazon S3 unterstützen bis zu 50 Tags für jeden Bucket. • Mit einem Bucket verknüpfte Tags müssen eindeutige Tag-Schlüssel haben. Ein Tag-Schlüssel kann bis zu 128 Unicode-Zeichen lang sein. • Tag-Werte können bis zu 256 Unicode-Zeichen lang sein. • Bei Schlüsseln und Werten wird zwischen Groß- und Kleinschreibung unterschieden. **Achtung**: Wenn für diesen Bucket ein nicht standardmäßiger ILM-Richtlinientag festgelegt ist, wird ein `NTAP-SG-ILM-BUCKET-TAG` Bucket-Tag mit einem ihm zugewiesenen Wert. Stellen Sie sicher, dass die `NTAP-SG-ILM-BUCKET-TAG` Das Bucket-Tag ist mit dem zugewiesenen Wert in allen PutBucketTagging-Anfragen enthalten. Ändern oder entfernen Sie dieses Tag nicht. **Hinweis**: Dieser Vorgang überschreibt alle aktuellen Tags, die der Bucket bereits hat. Wenn vorhandene Tags aus dem Set weggelassen werden, werden diese Tags für den Bucket entfernt.
PutBucketVersioning	Verwendet die `versioning` Unterressource zum Festlegen des Versionsstatus eines vorhandenen Buckets. Sie können den Versionsstatus mit einem der folgenden Werte festlegen: • Aktiviert: Aktiviert die Versionierung für die Objekte im Bucket. Alle dem Bucket hinzugefügten Objekte erhalten eine eindeutige Versions-ID. • Angehalten: Deaktiviert die Versionierung für die Objekte im Bucket. Alle zum Bucket hinzugefügten Objekte erhalten die Versions-ID `null` .
PutObjectLockConfiguration	Konfiguriert oder entfernt den Standardaufbewahrungsmodus und die Standardaufbewahrungsdauer des Buckets. Wenn die Standardaufbewahrungsfrist geändert wird, bleibt das Aufbewahrungsdatum vorhandener Objektversionen gleich und wird nicht anhand der neuen Standardaufbewahrungsfrist neu berechnet. Sehen"Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren" für detaillierte Informationen.

Operationen an Objekten

Operationen an Objekten

In diesem Abschnitt wird beschrieben, wie das StorageGRID -System S3 REST-API-Operationen für Objekte implementiert.

Für alle Objektoperationen gelten folgende Bedingungen:

- StorageGRID"Konsistenzwerte" werden von allen Operationen an Objekten unterstützt, mit Ausnahme der folgenden:
  - GetObjectAcl
  - `OPTIONS /`
  - PutObjectLegalHold
  - PutObjectRetention
  - SelectObjectContent
- 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.
- Alle Objekte in einem StorageGRID Bucket sind Eigentum des Bucket-Eigentümers, einschließlich der von einem anonymen Benutzer oder einem anderen Konto erstellten Objekte.
- Auf Datenobjekte, die über Swift in das StorageGRID System aufgenommen werden, kann nicht über S3 zugegriffen werden.

Die folgende Tabelle beschreibt, wie StorageGRID S3 REST API-Objektoperationen implementiert.

Betrieb	Durchführung
Objekt löschen	Multi-Faktor-Authentifizierung (MFA) und der Antwortheader `x-amz-mfa` werden nicht unterstützt. Bei der Verarbeitung einer DeleteObject-Anforderung versucht StorageGRID , alle Kopien des Objekts sofort von allen gespeicherten Standorten zu entfernen. Bei Erfolg gibt StorageGRID sofort eine Antwort an den Client zurück. Wenn nicht alle Kopien innerhalb von 30 Sekunden entfernt werden können (z. B. weil ein Speicherort vorübergehend nicht verfügbar ist), stellt StorageGRID die Kopien zur Entfernung in die Warteschlange und zeigt dem Client anschließend den Erfolg an. **Versionierung** Um eine bestimmte Version zu entfernen, muss der Anforderer der Bucket-Eigentümer sein und die `versionId` Unterressource. Durch die Verwendung dieser Unterressource wird die Version dauerhaft gelöscht. Wenn die `versionId` entspricht einem Löschmarker, der Antwortheader `x-amz-delete-marker` wird zurückgegeben auf `true` . • Wenn ein Objekt gelöscht wird, ohne dass `versionId` Unterressource auf einem Bucket mit aktivierter Versionierung, führt dies zur Generierung einer Löschmarkierung. Der `versionId` für die Löschmarkierung wird mit dem `x-amz-version-id` Antwortheader und der `x-amz-delete-marker` Der Antwortheader wird auf `true` . • Wenn ein Objekt gelöscht wird, ohne dass `versionId` Unterressource auf einem Bucket mit ausgesetzter Versionierung, führt dies zu einer dauerhaften Löschung einer bereits vorhandenen „Null“-Version oder eines „Null“-Löschmarkers und zur Generierung eines neuen „Null“-Löschmarkers. Der `x-amz-delete-marker` Der Antwortheader wird auf `true` . **Hinweis**: In bestimmten Fällen können für ein Objekt mehrere Löschmarkierungen vorhanden sein. Sehen"Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren" um zu erfahren, wie Sie Objektversionen im GOVERNANCE-Modus löschen.

Betrieb	Durchführung
Objekte löschen (früher „DELETE Multiple Objects“ genannt)	Multi-Faktor-Authentifizierung (MFA) und der Antwortheader `x-amz-mfa` werden nicht unterstützt. In derselben Anforderungsnachricht können mehrere Objekte gelöscht werden. Sehen"Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren" um zu erfahren, wie Sie Objektversionen im GOVERNANCE-Modus löschen.
DeleteObjectTagging	Verwendet die `tagging` Unterressource zum Entfernen aller Tags von einem Objekt. **Versionierung** Wenn die `versionId` Wenn in der Anforderung kein Abfrageparameter angegeben ist, löscht der Vorgang alle Tags aus der aktuellsten Version des Objekts in einem versionierten Bucket. Wenn die aktuelle Version des Objekts eine Löschmarkierung ist, wird der Status "MethodNotAllowed" mit der `x-amz-delete-marker` Antwortheader gesetzt auf `true` .
GetObject	"GetObject"
GetObjectAcl	Wenn die erforderlichen Zugriffsberechtigungen für das Konto bereitgestellt werden, gibt der Vorgang eine positive Antwort sowie die ID, den Anzeigenamen und die Berechtigung des Objektbesitzers zurück, was darauf hinweist, dass der Besitzer vollen Zugriff auf das Objekt hat.
GetObjectLegalHold	"Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren"
GetObjectRetention	"Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren"
GetObjectTagging	Verwendet die `tagging` Unterressource, um alle Tags für ein Objekt zurückzugeben. **Versionierung** Wenn die `versionId` Wenn in der Anforderung kein Abfrageparameter angegeben ist, gibt der Vorgang alle Tags aus der aktuellsten Version des Objekts in einem versionierten Bucket zurück. Wenn die aktuelle Version des Objekts eine Löschmarkierung ist, wird der Status "MethodNotAllowed" mit der `x-amz-delete-marker` Antwortheader gesetzt auf `true` .
HeadObject	"HeadObject"
RestoreObject	"RestoreObject"

Betrieb	Durchführung
PutObject	"PutObject"
Objekt kopieren (früher PUT-Objekt – Kopieren genannt)	"Objekt kopieren"
PutObjectLegalHold	"Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren"
PutObjectRetention	"Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren"

Betrieb	Durchführung
PutObjectTagging	Verwendet die `tagging` Unterressource zum Hinzufügen einer Reihe von Tags zu einem vorhandenen Objekt. **Objekt-Tag-Grenzwerte** Sie können neuen Objekten beim Hochladen Tags hinzufügen oder Sie können sie vorhandenen Objekten hinzufügen. Sowohl StorageGRID als auch Amazon S3 unterstützen bis zu 10 Tags für jedes Objekt. Mit einem Objekt verknüpfte Tags müssen eindeutige Tag-Schlüssel haben. Ein Tag-Schlüssel kann bis zu 128 Unicode-Zeichen lang sein und Tag-Werte können bis zu 256 Unicode-Zeichen lang sein. Bei Schlüsseln und Werten wird zwischen Groß- und Kleinschreibung unterschieden. **Tag-Updates und Aufnahmeverhalten** Wenn Sie PutObjectTagging verwenden, um die Tags eines Objekts zu aktualisieren, nimmt StorageGRID das Objekt nicht erneut auf. Dies bedeutet, dass die in der entsprechenden ILM-Regel angegebene Option für das Aufnahmeverhalten nicht verwendet wird. Alle durch die Aktualisierung ausgelösten Änderungen an der Objektplatzierung werden vorgenommen, wenn ILM durch normale ILM-Hintergrundprozesse neu ausgewertet wird. Dies bedeutet, dass keine Aktion ausgeführt wird, wenn die ILM-Regel die Option „Streng“ für das Aufnahmeverhalten verwendet und die erforderlichen Objektplatzierungen nicht vorgenommen werden können (z. B. weil ein neu erforderlicher Speicherort nicht verfügbar ist). Das aktualisierte Objekt behält seine aktuelle Platzierung bei, bis die erforderliche Platzierung möglich ist. **Konflikte lösen** 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. **Versionierung** Wenn die `versionId` Wenn in der Anforderung kein Abfrageparameter angegeben ist, fügt der Vorgang der aktuellsten Version des Objekts in einem versionierten Bucket Tags hinzu. Wenn die aktuelle Version des Objekts eine Löschmarkierung ist, wird der Status "MethodNotAllowed" mit der `x-amz-delete-marker` Antwortheader gesetzt auf `true` .
SelectObjectContent	"SelectObjectContent"

Verwenden Sie S3 Select

StorageGRID unterstützt die folgenden Amazon S3 Select-Klauseln, Datentypen und

Operatoren für die"SelectObjectContent-Befehl" .

Nicht aufgeführte Artikel werden nicht unterstützt.

Informationen zur Syntax finden Sie unter"SelectObjectContent" . Weitere Informationen zu S3 Select finden Sie im "AWS-Dokumentation für S3 Select" .

Nur Mandantenkonten, bei denen S3 Select aktiviert ist, können SelectObjectContent-Abfragen ausgeben. Siehe die"Überlegungen und Anforderungen zur Verwendung von S3 Select" .

Klauseln

- SELECT-Liste
- FROM-Klausel
- WHERE-Klausel
- LIMIT-Klausel

Datentypen

- bool
- ganze Zahl
- Schnur
- schweben
- Dezimal, numerisch
- Zeitstempel

Betreiber

Logische Operatoren

- UND
- NICHT
- ODER

Vergleichsoperatoren

- <
- >
- ⇐
- >=
- =
- =
- <>
- !=
- ZWISCHEN

- IN

Mustervergleichsoperatoren

- WIE
- _
- %

Unitäre Operatoren

- IST NULL
- IST NICHT NULL

Mathematische Operatoren

- +
- -
- *
- /
- %

StorageGRID folgt der Priorität des Amazon S3 Select-Operators.

Aggregatfunktionen

- AVG()
- ZÄHLEN(*)
- MAX()
- MIN()
- SUMME()

Bedingte Funktionen

- FALL
- VERSCHMELZEN
- NULLIF

Konvertierungsfunktionen

- CAST (für unterstützten Datentyp)

Datumsfunktionen

- DATE_ADD
- DATE_DIFF
- EXTRAKT
- TO_STRING

- TO_TIMESTAMP
- UTCNOW

Zeichenfolgenfunktionen

- CHAR_LENGTH, CHARACTER_LENGTH
- UNTERE
- TEILZEICHENKETTE
- TRIMMEN
- OBERE

Verwenden Sie serverseitige Verschlüsselung

Durch die serverseitige Verschlüsselung können Sie Ihre ruhenden Objektdaten schützen. StorageGRID verschlüsselt die Daten beim Schreiben des Objekts und entschlüsselt die Daten, wenn Sie auf das Objekt zugreifen.

Wenn Sie serverseitige Verschlüsselung verwenden möchten, können Sie je nach Verwaltung der Verschlüsselungsschlüssel zwischen zwei sich gegenseitig ausschließenden Optionen wählen:

- **SSE (serverseitige Verschlüsselung mit von StorageGRID verwalteten Schlüsseln)**: Wenn Sie eine S3-Anfrage zum Speichern eines Objekts stellen, verschlüsselt StorageGRID das Objekt mit einem eindeutigen Schlüssel. Wenn Sie eine S3-Anforderung zum Abrufen des Objekts stellen, verwendet StorageGRID den gespeicherten Schlüssel zum Entschlüsseln des Objekts.
- **SSE-C (serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln)**: Wenn Sie eine S3-Anfrage zum Speichern eines Objekts stellen, geben Sie Ihren eigenen Verschlüsselungsschlüssel an. Wenn Sie ein Objekt abrufen, geben Sie im Rahmen Ihrer Anfrage denselben Verschlüsselungsschlüssel an. Wenn die beiden Verschlüsselungsschlüssel übereinstimmen, wird das Objekt entschlüsselt und Ihre Objektdaten werden zurückgegeben.

  Während StorageGRID alle Objektverschlüsselungs- und -entschlüsselungsvorgänge verwaltet, müssen Sie die von Ihnen bereitgestellten Verschlüsselungsschlüssel verwalten.

> Die von Ihnen bereitgestellten Verschlüsselungsschlüssel werden niemals gespeichert. Wenn Sie einen Verschlüsselungsschlüssel verlieren, verlieren Sie das entsprechende Objekt.

> Wenn ein Objekt mit SSE oder SSE-C verschlüsselt ist, werden alle Verschlüsselungseinstellungen auf Bucket- oder Grid-Ebene ignoriert.

Verwenden Sie SSE

Um ein Objekt mit einem eindeutigen, von StorageGRID verwalteten Schlüssel zu verschlüsseln, verwenden Sie den folgenden Anforderungsheader:

```
x-amz-server-side-encryption
```

Der SSE-Anforderungsheader wird von den folgenden Objektoperationen unterstützt:

- "PutObject"
- "Objekt kopieren"
- "CreateMultipartUpload"

Verwenden Sie SSE-C

Um ein Objekt mit einem eindeutigen Schlüssel zu verschlüsseln, den Sie verwalten, verwenden Sie drei Anforderungsheader:

Anforderungsheader	Beschreibung
`x-amz-server-side-encryption-customer-algorithm`	Geben Sie den Verschlüsselungsalgorithmus an. Der Headerwert muss `AES256`.
`x-amz-server-side-encryption-customer-key`	Geben Sie den Verschlüsselungsschlüssel an, der zum Verschlüsseln oder Entschlüsseln des Objekts verwendet wird. Der Wert für den Schlüssel muss 256 Bit lang und base64-codiert sein.
`x-amz-server-side-encryption-customer-key-MD5`	Geben Sie den MD5-Digest des Verschlüsselungsschlüssels gemäß RFC 1321 an, der verwendet wird, um sicherzustellen, dass der Verschlüsselungsschlüssel fehlerfrei übertragen wurde. Der Wert für den MD5-Digest muss base64-codiert und 128 Bit lang sein.

Die SSE-C-Anforderungsheader werden von den folgenden Objektoperationen unterstützt:

- "GetObject"
- "HeadObject"
- "PutObject"
- "Objekt kopieren"
- "CreateMultipartUpload"
- "UploadPart"
- "UploadPartCopy"

Überlegungen zur Verwendung der serverseitigen Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C)

Beachten Sie vor der Verwendung von SSE-C die folgenden Hinweise:

- Sie müssen https verwenden.

> ⚠ StorageGRID lehnt bei Verwendung von SSE-C alle über HTTP gestellten Anfragen ab. Aus Sicherheitsgründen sollten Sie jeden versehentlich über HTTP gesendeten Schlüssel als gefährdet betrachten. Entsorgen Sie den Schlüssel und drehen Sie ihn entsprechend.

- Der ETag in der Antwort ist nicht der MD5 der Objektdaten.
- Sie müssen die Zuordnung von Verschlüsselungsschlüsseln zu Objekten verwalten. StorageGRID speichert keine Verschlüsselungsschlüssel. Sie sind für die Nachverfolgung des Verschlüsselungsschlüssels verantwortlich, den Sie für jedes Objekt bereitstellen.

- Wenn für Ihren Bucket die Versionierung aktiviert ist, sollte jede Objektversion über einen eigenen Verschlüsselungsschlüssel verfügen. Sie sind für die Nachverfolgung des für jede Objektversion verwendeten Verschlüsselungsschlüssels verantwortlich.
- Da Sie die Verschlüsselungsschlüssel auf der Clientseite verwalten, müssen Sie auch alle zusätzlichen Sicherheitsvorkehrungen, wie etwa die Schlüsselrotation, auf der Clientseite verwalten.

> Die von Ihnen bereitgestellten Verschlüsselungsschlüssel werden niemals gespeichert. Wenn Sie einen Verschlüsselungsschlüssel verlieren, verlieren Sie das entsprechende Objekt.

- Wenn für den Bucket eine Cross-Grid-Replikation oder eine CloudMirror-Replikation konfiguriert ist, können Sie keine SSE-C-Objekte aufnehmen. Der Aufnahmevorgang schlägt fehl.

Ähnliche Informationen

"Amazon S3-Benutzerhandbuch: Verwenden der serverseitigen Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C)"

Objekt kopieren

Mit der S3 CopyObject-Anforderung können Sie eine Kopie eines Objekts erstellen, das bereits in S3 gespeichert ist. Ein CopyObject-Vorgang ist dasselbe wie die Ausführung von GetObject gefolgt von PutObject.

Konflikte lösen

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.

Objektgröße

Die maximal *empfohlene* Größe für einen einzelnen PutObject-Vorgang beträgt 5 GiB (5.368.709.120 Bytes). Wenn Sie Objekte haben, die größer als 5 GiB sind, verwenden Sie"mehrteiliger Upload" stattdessen.

Die maximal *unterstützte* Größe für einen einzelnen PutObject-Vorgang beträgt 5 TiB (5.497.558.138.880 Bytes).

> Wenn Sie ein Upgrade von StorageGRID 11.6 oder früher durchgeführt haben, wird die Warnung „S3 PUT-Objektgröße zu groß“ ausgelöst, wenn Sie versuchen, ein Objekt hochzuladen, das 5 GiB überschreitet. Wenn Sie eine Neuinstallation von StorageGRID 11.7 oder 11.8 haben, wird der Alarm in diesem Fall nicht ausgelöst. Um jedoch dem AWS S3-Standard zu entsprechen, werden zukünftige Versionen von StorageGRID keine Uploads von Objekten unterstützen, die größer als 5 GiB sind.

UTF-8-Zeichen in Benutzermetadaten

Wenn eine Anfrage (nicht maskierte) UTF-8-Werte im Schlüsselnamen oder Wert benutzerdefinierter Metadaten enthält, ist das StorageGRID Verhalten undefiniert.

StorageGRID analysiert oder interpretiert keine Escape-UTF-8-Zeichen, die im Schlüsselnamen oder -wert benutzerdefinierter Metadaten enthalten sind. Escape-UTF-8-Zeichen werden als ASCII-Zeichen behandelt:

- Anforderungen sind erfolgreich, wenn benutzerdefinierte Metadaten Escape-UTF-8-Zeichen enthalten.
- StorageGRID gibt nicht zurück `x-amz-missing-meta` Header, wenn der interpretierte Wert des Schlüsselnamens oder -werts nicht druckbare Zeichen enthält.

Unterstützte Anforderungsheader

Die folgenden Anforderungsheader werden unterstützt:

- `Content-Type`
- `x-amz-copy-source`
- `x-amz-copy-source-if-match`
- `x-amz-copy-source-if-none-match`
- `x-amz-copy-source-if-unmodified-since`
- `x-amz-copy-source-if-modified-since`
- `x-amz-meta-`, gefolgt von einem Name-Wert-Paar mit benutzerdefinierten Metadaten
- `x-amz-metadata-directive`: Der Standardwert ist `COPY` , wodurch Sie das Objekt und die zugehörigen Metadaten kopieren können.

  Sie können angeben `REPLACE` um die vorhandenen Metadaten beim Kopieren des Objekts zu überschreiben oder die Objektmetadaten zu aktualisieren.

- `x-amz-storage-class`
- `x-amz-tagging-directive`: Der Standardwert ist `COPY` , wodurch Sie das Objekt und alle Tags kopieren können.

  Sie können angeben `REPLACE` um die vorhandenen Tags beim Kopieren des Objekts zu überschreiben oder die Tags zu aktualisieren.

- S3 Object Lock-Anforderungsheader:
  - `x-amz-object-lock-mode`
  - `x-amz-object-lock-retain-until-date`
  - `x-amz-object-lock-legal-hold`

    Wenn eine Anfrage ohne diese Header gestellt wird, werden die Bucket-Standardaufbewahrungseinstellungen verwendet, um den Objektversionsmodus und das Aufbewahrungsdatum zu berechnen. Sehen "Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren" .

- SSE-Anforderungsheader:
  - `x-amz-copy-source-server-side-encryption-customer-algorithm`
  - `x-amz-copy-source-server-side-encryption-customer-key`
  - `x-amz-copy-source-server-side-encryption-customer-key-MD5`
  - `x-amz-server-side-encryption`
  - `x-amz-server-side-encryption-customer-key-MD5`

- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-algorithm`

  SehenAnforderungsheader für serverseitige Verschlüsselung

Nicht unterstützte Anforderungsheader

Die folgenden Anforderungsheader werden nicht unterstützt:

- `Cache-Control`
- `Content-Disposition`
- `Content-Encoding`
- `Content-Language`
- `Expires`
- `x-amz-checksum-algorithm`

  Wenn Sie ein Objekt kopieren und das Quellobjekt eine Prüfsumme hat, kopiert StorageGRID diesen Prüfsummenwert nicht in das neue Objekt. Dieses Verhalten gilt unabhängig davon, ob Sie versuchen, `x-amz-checksum-algorithm` in der Objektanforderung.

- `x-amz-website-redirect-location`

Speicherklassenoptionen

Der `x-amz-storage-class` Der Anforderungsheader wird unterstützt und beeinflusst, wie viele Objektkopien StorageGRID erstellt, wenn die entsprechende ILM-Regel Dual Commit oder Balanced verwendet."Aufnahmeoption" .

- `STANDARD`

  (Standard) Gibt einen Dual-Commit-Aufnahmevorgang an, wenn die ILM-Regel die Option „Dual Commit“ verwendet oder wenn die Option „Balanced“ auf die Erstellung von Zwischenkopien zurückgreift.

- `REDUCED_REDUNDANCY`

  Gibt einen Single-Commit-Ingest-Vorgang an, wenn die ILM-Regel die Option „Dual Commit“ verwendet oder wenn die Option „Balanced“ auf die Erstellung von Zwischenkopien zurückgreift.

  > Wenn Sie ein Objekt in einen Bucket mit aktivierter S3-Objektsperre aufnehmen, wird die `REDUCED_REDUNDANCY` Option wird ignoriert. Wenn Sie ein Objekt in einen Legacy-Compliant-Bucket aufnehmen, `REDUCED_REDUNDANCY` Option gibt einen Fehler zurück. StorageGRID führt immer eine Dual-Commit-Aufnahme durch, um sicherzustellen, dass die Compliance-Anforderungen erfüllt werden.

Verwenden von x-amz-copy-source in CopyObject

Wenn der Quell-Bucket und -Schlüssel, angegeben in `x-amz-copy-source` Header, unterscheiden sich vom Ziel-Bucket und -Schlüssel, eine Kopie der Quellobjektdaten wird in das Ziel geschrieben.

Wenn Quelle und Ziel übereinstimmen und die `x-amz-metadata-directive` Der Header wird wie folgt angegeben: `REPLACE` , werden die Metadaten des Objekts mit den in der Anfrage angegebenen Metadatenwerten aktualisiert. In diesem Fall nimmt StorageGRID das Objekt nicht erneut auf. Dies hat zwei wichtige Konsequenzen:

- Sie können CopyObject nicht verwenden, um ein vorhandenes Objekt vor Ort zu verschlüsseln oder die Verschlüsselung eines vorhandenen Objekts vor Ort zu ändern. Wenn Sie die `x-amz-server-side-encryption` Kopfzeile oder die `x-amz-server-side-encryption-customer-algorithm` Header, StorageGRID lehnt die Anfrage ab und gibt zurück `XNotImplemented` .
- Die in der entsprechenden ILM-Regel angegebene Option für das Aufnahmeverhalten wird nicht verwendet. Alle durch die Aktualisierung ausgelösten Änderungen an der Objektplatzierung werden vorgenommen, wenn ILM durch normale ILM-Hintergrundprozesse neu ausgewertet wird.

  Dies bedeutet, dass keine Aktion ausgeführt wird, wenn die ILM-Regel die Option „Streng“ für das Aufnahmeverhalten verwendet und die erforderlichen Objektplatzierungen nicht vorgenommen werden können (z. B. weil ein neu erforderlicher Speicherort nicht verfügbar ist). Das aktualisierte Objekt behält seine aktuelle Platzierung bei, bis die erforderliche Platzierung möglich ist.

Anforderungsheader für serverseitige Verschlüsselung

Wenn du"Verwenden Sie serverseitige Verschlüsselung" , die von Ihnen bereitgestellten Anforderungsheader hängen davon ab, ob das Quellobjekt verschlüsselt ist und ob Sie das Zielobjekt verschlüsseln möchten.

- Wenn das Quellobjekt mit einem vom Kunden bereitgestellten Schlüssel (SSE-C) verschlüsselt ist, müssen Sie die folgenden drei Header in die CopyObject-Anforderung aufnehmen, damit das Objekt entschlüsselt und dann kopiert werden kann:
  - `x-amz-copy-source-server-side-encryption-customer-algorithm`: Angeben `AES256` .
  - `x-amz-copy-source-server-side-encryption-customer-key`: Geben Sie den Verschlüsselungsschlüssel an, den Sie beim Erstellen des Quellobjekts angegeben haben.
  - `x-amz-copy-source-server-side-encryption-customer-key-MD5`: Geben Sie den MD5-Digest an, den Sie beim Erstellen des Quellobjekts angegeben haben.
- Wenn Sie das Zielobjekt (die Kopie) mit einem eindeutigen Schlüssel verschlüsseln möchten, den Sie bereitstellen und verwalten, schließen Sie die folgenden drei Header ein:
  - `x-amz-server-side-encryption-customer-algorithm`: Angeben `AES256` .
  - `x-amz-server-side-encryption-customer-key`: Geben Sie einen neuen Verschlüsselungsschlüssel für das Zielobjekt an.
  - `x-amz-server-side-encryption-customer-key-MD5`: Geben Sie den MD5-Digest des neuen Verschlüsselungsschlüssels an.

  > Die von Ihnen bereitgestellten Verschlüsselungsschlüssel werden niemals gespeichert. Wenn Sie einen Verschlüsselungsschlüssel verlieren, verlieren Sie das entsprechende Objekt. Bevor Sie vom Kunden bereitgestellte Schlüssel zum Sichern von Objektdaten verwenden, lesen Sie die Überlegungen für"Verwendung serverseitiger Verschlüsselung" .
- Wenn Sie das Zielobjekt (die Kopie) mit einem eindeutigen, von StorageGRID (SSE) verwalteten Schlüssel verschlüsseln möchten, fügen Sie diesen Header in die CopyObject-Anforderung ein:
  - `x-amz-server-side-encryption`

Der `server-side-encryption` Der Wert des Objekts kann nicht aktualisiert werden. Erstellen Sie stattdessen eine Kopie mit einem neuen `server-side-encryption` Wert mit `x-amz-metadata-directive` : `REPLACE` .

Versionierung

Wenn der Quell-Bucket versioniert ist, können Sie die `x-amz-copy-source` Header, um die neueste Version eines Objekts zu kopieren. Um eine bestimmte Version eines Objekts zu kopieren, müssen Sie die zu kopierende Version explizit angeben, indem Sie `versionId` Unterressource. Wenn der Ziel-Bucket versioniert ist, wird die generierte Version im `x-amz-version-id` Antwortheader. Wenn die Versionierung für den Ziel-Bucket ausgesetzt ist, dann `x-amz-version-id` gibt einen „Null“-Wert zurück.

GetObject

Sie können die S3 GetObject-Anforderung verwenden, um ein Objekt aus einem S3-Bucket abzurufen.

GetObject und mehrteilige Objekte

Sie können die `partNumber` Anforderungsparameter zum Abrufen eines bestimmten Teils eines mehrteiligen oder segmentierten Objekts. Der `x-amz-mp-parts-count` Das Antwortelement gibt an, aus wie vielen Teilen das Objekt besteht.

Sie können einstellen `partNumber` auf 1 für segmentierte/mehrteilige Objekte und nicht-segmentierte/nicht-mehrteilige Objekte; jedoch `x-amz-mp-parts-count` Das Antwortelement wird nur für segmentierte oder mehrteilige Objekte zurückgegeben.

UTF-8-Zeichen in Benutzermetadaten

StorageGRID analysiert oder interpretiert keine Escape-UTF-8-Zeichen in benutzerdefinierten Metadaten. GET-Anfragen für ein Objekt mit Escape-UTF-8-Zeichen in benutzerdefinierten Metadaten geben nicht die `x-amz-missing-meta` Header, wenn der Schlüsselname oder -wert nicht druckbare Zeichen enthält.

Unterstützter Anforderungsheader

Der folgende Anforderungsheader wird unterstützt:

- `x-amz-checksum-mode`: Angeben `ENABLED`

  Der `Range` Header wird nicht unterstützt mit `x-amz-checksum-mode` für GetObject. Wenn Sie `Range` in der Anfrage mit `x-amz-checksum-mode` aktiviert ist, gibt StorageGRID in der Antwort keinen Prüfsummenwert zurück.

Nicht unterstützter Anforderungsheader

Der folgende Anforderungsheader wird nicht unterstützt und gibt zurück `XNotImplemented` :

- `x-amz-website-redirect-location`

Versionierung

Wenn ein `versionId` Wenn keine Unterressource angegeben ist, ruft der Vorgang die aktuellste Version des Objekts in einem versionierten Bucket ab. Wenn die aktuelle Version des Objekts eine Löschmarkierung ist, wird der Status "Nicht gefunden" mit der `x-amz-delete-marker` Antwortheader gesetzt auf `true` .

Anforderungsheader für die serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C)

Verwenden Sie alle drei Header, wenn das Objekt mit einem von Ihnen bereitgestellten eindeutigen Schlüssel verschlüsselt ist.

- `x-amz-server-side-encryption-customer-algorithm`: Angeben `AES256` .
- `x-amz-server-side-encryption-customer-key`: Geben Sie Ihren Verschlüsselungsschlüssel für das Objekt an.
- `x-amz-server-side-encryption-customer-key-MD5`: Geben Sie den MD5-Digest des Verschlüsselungsschlüssels des Objekts an.

> Die von Ihnen bereitgestellten Verschlüsselungsschlüssel werden niemals gespeichert. Wenn Sie einen Verschlüsselungsschlüssel verlieren, verlieren Sie das entsprechende Objekt. Bevor Sie vom Kunden bereitgestellte Schlüssel zum Sichern von Objektdaten verwenden, lesen Sie die Hinweise in"Verwenden Sie serverseitige Verschlüsselung" .

Verhalten von GetObject für Cloud Storage Pool-Objekte

Wenn ein Objekt in einem"Cloud-Speicherpool" , das Verhalten einer GetObject-Anforderung hängt vom Status des Objekts ab. Sehen"HeadObject" für weitere Details.

> Wenn ein Objekt in einem Cloud-Speicherpool gespeichert ist und eine oder mehrere Kopien des Objekts auch im Raster vorhanden sind, versuchen GetObject-Anfragen, Daten aus dem Raster abzurufen, bevor sie aus dem Cloud-Speicherpool abgerufen werden.

Zustand des Objekts	Verhalten von GetObject
In StorageGRID aufgenommenes, aber noch nicht von ILM ausgewertetes Objekt oder Objekt, das in einem herkömmlichen Speicherpool oder mithilfe von Erasure Coding gespeichert ist	`200 OK` Eine Kopie des Objekts wird abgerufen.
Objekt im Cloud-Speicherpool, aber noch nicht in einen nicht abrufbaren Zustand übergegangen	`200 OK` Eine Kopie des Objekts wird abgerufen.
Objekt in einen nicht abrufbaren Zustand überführt	`403 Forbidden` , `InvalidObjectState` Verwenden Sie ein"RestoreObject" Anforderung zum Wiederherstellen des Objekts in einen abrufbaren Zustand.

Zustand des Objekts	Verhalten von GetObject
Objekt wird gerade aus einem nicht abrufbaren Zustand wiederhergestellt	`403 Forbidden` , `InvalidObjectState` Warten Sie, bis die RestoreObject-Anforderung abgeschlossen ist.
Objekt vollständig im Cloud-Speicherpool wiederhergestellt	`200 OK` Eine Kopie des Objekts wird abgerufen.

Mehrteilige oder segmentierte Objekte in einem Cloud-Speicherpool

Wenn Sie ein mehrteiliges Objekt hochgeladen haben oder StorageGRID ein großes Objekt in Segmente aufgeteilt hat, ermittelt StorageGRID , ob das Objekt im Cloud Storage Pool verfügbar ist, indem es eine Teilmenge der Teile oder Segmente des Objekts auswählt. In einigen Fällen kann eine GetObject-Anforderung fälschlicherweise zurückgeben `200 OK` wenn einige Teile des Objekts bereits in einen nicht abrufbaren Zustand überführt wurden oder wenn einige Teile des Objekts noch nicht wiederhergestellt wurden.

In diesen Fällen:

- Die GetObject-Anforderung gibt möglicherweise einige Daten zurück, stoppt jedoch mitten in der Übertragung.
- Eine nachfolgende GetObject-Anforderung könnte `403 Forbidden` .

GetObject und Cross-Grid-Replikation

Wenn Sie"Netzverbund" Und"Cross-Grid-Replikation" für einen Bucket aktiviert ist, kann der S3-Client den Replikationsstatus eines Objekts überprüfen, indem er eine GetObject-Anforderung ausgibt. Die Antwort enthält die StorageGRID-spezifischen `x-ntap-sg-cgr-replication-status` Antwortheader, der einen der folgenden Werte hat:

Netz	Replikationsstatus
Quelle	• **ABGESCHLOSSEN**: Die Replikation war erfolgreich. • **AUSSTEHEND**: Das Objekt wurde noch nicht repliziert. • **FEHLER**: Die Replikation ist mit einem dauerhaften Fehler fehlgeschlagen. Der Fehler muss von einem Benutzer behoben werden.
Ziel	**REPLICA**: Das Objekt wurde aus dem Quellraster repliziert.

StorageGRID unterstützt nicht die `x-amz-replication-status` Kopfzeile.

HeadObject

Sie können die S3 HeadObject-Anforderung verwenden, um Metadaten aus einem Objekt abzurufen, ohne das Objekt selbst zurückzugeben. Wenn das Objekt in einem Cloud-Speicherpool gespeichert ist, können Sie HeadObject verwenden, um den

Übergangszustand des Objekts zu bestimmen.

HeadObject und mehrteilige Objekte

Sie können die `partNumber` Anforderungsparameter zum Abrufen von Metadaten für einen bestimmten Teil eines mehrteiligen oder segmentierten Objekts. Der `x-amz-mp-parts-count` Das Antwortelement gibt an, aus wie vielen Teilen das Objekt besteht.

Sie können einstellen `partNumber` auf 1 für segmentierte/mehrteilige Objekte und nicht-segmentierte/nicht-mehrteilige Objekte; jedoch `x-amz-mp-parts-count` Das Antwortelement wird nur für segmentierte oder mehrteilige Objekte zurückgegeben.

UTF-8-Zeichen in Benutzermetadaten

StorageGRID analysiert oder interpretiert keine Escape-UTF-8-Zeichen in benutzerdefinierten Metadaten. HEAD-Anfragen für ein Objekt mit Escape-UTF-8-Zeichen in benutzerdefinierten Metadaten geben nicht das `x-amz-missing-meta` Header, wenn der Schlüsselname oder -wert nicht druckbare Zeichen enthält.

Unterstützter Anforderungsheader

Der folgende Anforderungsheader wird unterstützt:

- `x-amz-checksum-mode`

  Der `partNumber` Parameter und `Range` Header werden nicht unterstützt mit `x-amz-checksum-mode` für HeadObject. Wenn Sie sie in die Anfrage aufnehmen mit `x-amz-checksum-mode` aktiviert ist, gibt StorageGRID in der Antwort keinen Prüfsummenwert zurück.

Nicht unterstützter Anforderungsheader

Der folgende Anforderungsheader wird nicht unterstützt und gibt zurück `XNotImplemented` :

- `x-amz-website-redirect-location`

Versionierung

Wenn ein `versionId` Wenn keine Unterressource angegeben ist, ruft der Vorgang die aktuellste Version des Objekts in einem versionierten Bucket ab. Wenn die aktuelle Version des Objekts eine Löschmarkierung ist, wird der Status "Nicht gefunden" mit der `x-amz-delete-marker` Antwortheader gesetzt auf `true` .

Anforderungsheader für die serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C)

Verwenden Sie alle drei Header, wenn das Objekt mit einem von Ihnen bereitgestellten eindeutigen Schlüssel verschlüsselt ist.

- `x-amz-server-side-encryption-customer-algorithm`: Angeben `AES256` .
- `x-amz-server-side-encryption-customer-key`: Geben Sie Ihren Verschlüsselungsschlüssel für das Objekt an.
- `x-amz-server-side-encryption-customer-key-MD5`: Geben Sie den MD5-Digest des Verschlüsselungsschlüssels des Objekts an.

Die von Ihnen bereitgestellten Verschlüsselungsschlüssel werden niemals gespeichert. Wenn Sie einen Verschlüsselungsschlüssel verlieren, verlieren Sie das entsprechende Objekt. Bevor Sie vom Kunden bereitgestellte Schlüssel zum Sichern von Objektdaten verwenden, lesen Sie die Hinweise in"Verwenden Sie serverseitige Verschlüsselung" .

HeadObject-Antworten für Cloud Storage Pool-Objekte

Wenn das Objekt in einem"Cloud-Speicherpool" werden die folgenden Antwortheader zurückgegeben:

- `x-amz-storage-class: GLACIER`
- `x-amz-restore`

Die Antwortheader liefern Informationen über den Status eines Objekts, wenn es in einen Cloud-Speicherpool verschoben, optional in einen nicht abrufbaren Status versetzt und wiederhergestellt wird.

Zustand des Objekts	Antwort auf HeadObject
In StorageGRID aufgenommenes, aber noch nicht von ILM ausgewertetes Objekt oder Objekt, das in einem herkömmlichen Speicherpool oder mithilfe von Erasure Coding gespeichert ist	`200 OK`(Es wird kein spezieller Antwortheader zurückgegeben.)

Objekt im Cloud-Speicherpool, aber noch nicht in einen nicht abrufbaren Zustand übergegangen	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="false", expiry-date="Sat, 23 July 20 2030 00:00:00 GMT"`
	
	Bis das Objekt in einen nicht abrufbaren Zustand überführt wird, ist der Wert für `expiry-date` ist auf einen fernen Zeitpunkt in der Zukunft festgelegt. Der genaue Zeitpunkt des Übergangs wird vom StorageGRID -System nicht gesteuert.

Zustand des Objekts	Antwort auf HeadObject

Das Objekt ist in den nicht abrufbaren Zustand übergegangen, aber mindestens eine Kopie ist auch im Raster vorhanden	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="false", expiry-date="Sat, 23 July 20 2030 00:00:00 GMT"`
	
	Der Wert für `expiry-date` ist auf einen fernen Zeitpunkt in der Zukunft festgelegt.
	
	Hinweis: Wenn die Kopie im Grid nicht verfügbar ist (z. B. weil ein Storage Node ausgefallen ist), müssen Sie eine"RestoreObject" Fordern Sie die Wiederherstellung der Kopie aus dem Cloud-Speicherpool an, bevor Sie das Objekt erfolgreich abrufen können.

Das Objekt ist in einen nicht abrufbaren Zustand übergegangen und es ist keine Kopie im Raster vorhanden	`200 OK`
	
	`x-amz-storage-class: GLACIER`

Objekt wird gerade aus einem nicht abrufbaren Zustand wiederhergestellt	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="true"`

Objekt vollständig im Cloud-Speicherpool wiederhergestellt	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="false", expiry-date="Sat, 23 July 20 2018 00:00:00 GMT"`
	
	Der `expiry-date` gibt an, wann das Objekt im Cloud-Speicherpool in einen nicht abrufbaren Zustand zurückversetzt wird.

Mehrteilige oder segmentierte Objekte im Cloud Storage Pool

Wenn Sie ein mehrteiliges Objekt hochgeladen haben oder StorageGRID ein großes Objekt in Segmente aufgeteilt hat, ermittelt StorageGRID , ob das Objekt im Cloud Storage Pool verfügbar ist, indem es eine Teilmenge der Teile oder Segmente des Objekts auswählt. In einigen Fällen kann eine HeadObject-Anforderung fälschlicherweise zurückgeben `x-amz-restore: ongoing-request="false"` wenn einige Teile des Objekts bereits in einen nicht abrufbaren Zustand überführt wurden oder wenn einige Teile des Objekts noch nicht wiederhergestellt wurden.

HeadObject und Cross-Grid-Replikation

Wenn Sie"Netzverbund" Und"Cross-Grid-Replikation" für einen Bucket aktiviert ist, kann der S3-Client den Replikationsstatus eines Objekts überprüfen, indem er eine HeadObject-Anforderung ausgibt. Die Antwort enthält die StorageGRID-spezifischen `x-ntap-sg-cgr-replication-status` Antwortheader, der einen der folgenden Werte hat:

Netz	Replikationsstatus
Quelle	• **ABGESCHLOSSEN**: Die Replikation war erfolgreich. • **AUSSTEHEND**: Das Objekt wurde noch nicht repliziert. • **FEHLER**: Die Replikation ist mit einem dauerhaften Fehler fehlgeschlagen. Der Fehler muss von einem Benutzer behoben werden.
Ziel	**REPLICA**: Das Objekt wurde aus dem Quellraster repliziert.

StorageGRID unterstützt nicht die `x-amz-replication-status` Kopfzeile.

PutObject

Sie können die S3 PutObject-Anforderung verwenden, um einem Bucket ein Objekt hinzuzufügen.

Konflikte lösen

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.

Objektgröße

Die maximal *empfohlene* Größe für einen einzelnen PutObject-Vorgang beträgt 5 GiB (5.368.709.120 Bytes). Wenn Sie Objekte haben, die größer als 5 GiB sind, verwenden Sie"mehrteiliger Upload" stattdessen.

Die maximal *unterstützte* Größe für einen einzelnen PutObject-Vorgang beträgt 5 TiB (5.497.558.138.880 Bytes).

Wenn Sie ein Upgrade von StorageGRID 11.6 oder früher durchgeführt haben, wird die Warnung „S3 PUT-Objektgröße zu groß“ ausgelöst, wenn Sie versuchen, ein Objekt hochzuladen, das 5 GiB überschreitet. Wenn Sie eine Neuinstallation von StorageGRID 11.7 oder 11.8 haben, wird der Alarm in diesem Fall nicht ausgelöst. Um jedoch dem AWS S3-Standard zu entsprechen, werden zukünftige Versionen von StorageGRID keine Uploads von Objekten unterstützen, die größer als 5 GiB sind.

Größe der Benutzermetadaten

Amazon S3 begrenzt die Größe benutzerdefinierter Metadaten innerhalb jedes PUT-Anforderungsheaders auf 2 KB. StorageGRID begrenzt Benutzermetadaten auf 24 KiB. Die Größe benutzerdefinierter Metadaten wird gemessen, indem die Summe der Anzahl der Bytes in der UTF-8-Kodierung jedes Schlüssels und Werts

berechnet wird.

UTF-8-Zeichen in Benutzermetadaten

Wenn eine Anfrage (nicht maskierte) UTF-8-Werte im Schlüsselnamen oder Wert benutzerdefinierter Metadaten enthält, ist das StorageGRID Verhalten undefiniert.

StorageGRID analysiert oder interpretiert keine Escape-UTF-8-Zeichen, die im Schlüsselnamen oder -wert benutzerdefinierter Metadaten enthalten sind. Escape-UTF-8-Zeichen werden als ASCII-Zeichen behandelt:

- PutObject-, CopyObject-, GetObject- und HeadObject-Anfragen sind erfolgreich, wenn benutzerdefinierte Metadaten Escape-UTF-8-Zeichen enthalten.
- StorageGRID gibt nicht zurück `x-amz-missing-meta` Header, wenn der interpretierte Wert des Schlüsselnamens oder -werts nicht druckbare Zeichen enthält.

Objekt-Tag-Grenzwerte

Sie können neuen Objekten beim Hochladen Tags hinzufügen oder Sie können sie vorhandenen Objekten hinzufügen. Sowohl StorageGRID als auch Amazon S3 unterstützen bis zu 10 Tags für jedes Objekt. Mit einem Objekt verknüpfte Tags müssen eindeutige Tag-Schlüssel haben. Ein Tag-Schlüssel kann bis zu 128 Unicode-Zeichen lang sein und Tag-Werte können bis zu 256 Unicode-Zeichen lang sein. Bei Schlüsseln und Werten wird zwischen Groß- und Kleinschreibung unterschieden.

Objektbesitz

In StorageGRID sind alle Objekte Eigentum des Bucket-Eigentümerkontos, einschließlich der Objekte, die von einem Nicht-Eigentümerkonto oder einem anonymen Benutzer erstellt wurden.

Unterstützte Anforderungsheader

Die folgenden Anforderungsheader werden unterstützt:

- `Cache-Control`
- `Content-Disposition`
- `Content-Encoding`

  Wenn Sie angeben `aws-chunked` für`Content-Encoding` StorageGRID überprüft die folgenden Punkte nicht:

  - StorageGRID überprüft nicht die `chunk-signature` gegen die Chunk-Daten.
  - StorageGRID überprüft den Wert, den Sie angeben, nicht für `x-amz-decoded-content-length` gegen das Objekt.
- `Content-Language`
- `Content-Length`
- `Content-MD5`
- `Content-Type`
- `Expires`
- `Transfer-Encoding`

Chunked Transfer Encoding wird unterstützt, wenn `aws-chunked` Außerdem wird eine Nutzlastsignatur verwendet.

- `x-amz-checksum-sha256`
- `x-amz-meta-`, gefolgt von einem Name-Wert-Paar mit benutzerdefinierten Metadaten.

  Verwenden Sie beim Angeben des Name-Wert-Paares für benutzerdefinierte Metadaten dieses allgemeine Format:

  ```
  x-amz-meta-name: value
  ```

  Wenn Sie die Option **Benutzerdefinierte Erstellungszeit** als Referenzzeit für eine ILM-Regel verwenden möchten, müssen Sie `creation-time` als Name der Metadaten, die aufzeichnen, wann das Objekt erstellt wurde. Beispiel:

  ```
  x-amz-meta-creation-time: 1443399726
  ```

  Der Wert für `creation-time` wird seit dem 1. Januar 1970 in Sekunden ausgewertet.

  > Eine ILM-Regel kann nicht sowohl eine **benutzerdefinierte Erstellungszeit** für die Referenzzeit als auch die Aufnahmeoption „Ausgewogen“ oder „Streng“ verwenden. Beim Erstellen der ILM-Regel wird ein Fehler zurückgegeben.

- `x-amz-tagging`
- S3 Object Lock-Anforderungsheader
  - `x-amz-object-lock-mode`
  - `x-amz-object-lock-retain-until-date`
  - `x-amz-object-lock-legal-hold`

    Wenn eine Anfrage ohne diese Header gestellt wird, werden die Bucket-Standardaufbewahrungseinstellungen verwendet, um den Objektversionsmodus und das Aufbewahrungsdatum zu berechnen. Sehen "Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren" .

- SSE-Anforderungsheader:
  - `x-amz-server-side-encryption`
  - `x-amz-server-side-encryption-customer-key-MD5`
  - `x-amz-server-side-encryption-customer-key`
  - `x-amz-server-side-encryption-customer-algorithm`

    SehenAnforderungsheader für serverseitige Verschlüsselung

Nicht unterstützte Anforderungsheader

Die folgenden Anforderungsheader werden nicht unterstützt:

- `x-amz-acl`
- `x-amz-sdk-checksum-algorithm`
- `x-amz-trailer`
- `x-amz-website-redirect-location`

  Der `x-amz-website-redirect-location` Header-Returns `XNotImplemented` .

Speicherklassenoptionen

Der `x-amz-storage-class` Anforderungsheader wird unterstützt. Der übermittelte Wert für `x-amz-storage-class` beeinflusst, wie StorageGRID Objektdaten während der Aufnahme schützt, und nicht, wie viele persistente Kopien des Objekts im StorageGRID -System gespeichert werden (was durch ILM bestimmt wird).

Wenn die ILM-Regel, die einem aufgenommenen Objekt entspricht, die Option „Strenge Aufnahme“ verwendet, `x-amz-storage-class` Header hat keine Wirkung.

Folgende Werte können verwendet werden für `x-amz-storage-class` :

- `STANDARD`(Standard)
  - **Dual Commit**: Wenn die ILM-Regel die Option „Dual Commit“ für das Aufnahmeverhalten angibt, wird, sobald ein Objekt aufgenommen wird, eine zweite Kopie dieses Objekts erstellt und an einen anderen Speicherknoten verteilt (Dual Commit). Bei der Auswertung des ILM ermittelt StorageGRID , ob diese ersten Zwischenkopien die Platzierungsanweisungen in der Regel erfüllen. Ist dies nicht der Fall, müssen möglicherweise neue Objektkopien an anderen Orten erstellt und die ersten Zwischenkopien gelöscht werden.
  - **Ausgeglichen**: Wenn die ILM-Regel die Option „Ausgeglichen“ angibt und StorageGRID nicht sofort alle in der Regel angegebenen Kopien erstellen kann, erstellt StorageGRID zwei Zwischenkopien auf verschiedenen Speicherknoten.

    Wenn StorageGRID alle in der ILM-Regel angegebenen Objektkopien sofort erstellen kann (synchrone Platzierung), `x-amz-storage-class` Header hat keine Wirkung.

- `REDUCED_REDUNDANCY`
  - **Dual Commit**: Wenn die ILM-Regel die Option „Dual Commit“ für das Aufnahmeverhalten angibt, erstellt StorageGRID beim Aufnehmen des Objekts eine einzelne Zwischenkopie (Single Commit).
  - **Ausgeglichen**: Wenn die ILM-Regel die Option „Ausgeglichen“ angibt, erstellt StorageGRID nur dann eine einzelne Zwischenkopie, wenn das System nicht sofort alle in der Regel angegebenen Kopien erstellen kann. Wenn StorageGRID eine synchrone Platzierung durchführen kann, hat dieser Header keine Wirkung. Der `REDUCED_REDUNDANCY` Die Option wird am besten verwendet, wenn die ILM-Regel, die dem Objekt entspricht, eine einzelne replizierte Kopie erstellt. In diesem Fall mit `REDUCED_REDUNDANCY` vermeidet das unnötige Erstellen und Löschen einer zusätzlichen Objektkopie für jeden Aufnahmevorgang.

  Verwenden des `REDUCED_REDUNDANCY` Unter anderen Umständen wird diese Option nicht empfohlen. `REDUCED_REDUNDANCY` erhöht das Risiko eines Objektdatenverlusts während der Aufnahme.

Beispielsweise können Daten verloren gehen, wenn die einzelne Kopie zunächst auf einem Speicherknoten gespeichert wird, der ausfällt, bevor die ILM-Auswertung erfolgen kann.

> Wenn für einen bestimmten Zeitraum nur eine Kopie vorhanden ist, besteht die Gefahr eines dauerhaften Datenverlusts. Wenn nur eine replizierte Kopie eines Objekts vorhanden ist, geht dieses Objekt verloren, wenn ein Speicherknoten ausfällt oder einen schwerwiegenden Fehler aufweist. Auch während Wartungsvorgängen wie Upgrades verlieren Sie vorübergehend den Zugriff auf das Objekt.

Festlegen `REDUCED_REDUNDANCY` wirkt sich nur darauf aus, wie viele Kopien erstellt werden, wenn ein Objekt zum ersten Mal aufgenommen wird. Es hat keinen Einfluss darauf, wie viele Kopien des Objekts erstellt werden, wenn das Objekt von den aktiven ILM-Richtlinien ausgewertet wird, und führt nicht dazu, dass Daten im StorageGRID System auf niedrigeren Redundanzebenen gespeichert werden.

> Wenn Sie ein Objekt in einen Bucket mit aktivierter S3-Objektsperre aufnehmen, wird die `REDUCED_REDUNDANCY` Option wird ignoriert. Wenn Sie ein Objekt in einen Legacy-Compliant-Bucket aufnehmen, `REDUCED_REDUNDANCY` Option gibt einen Fehler zurück. StorageGRID führt immer eine Dual-Commit-Aufnahme durch, um sicherzustellen, dass die Compliance-Anforderungen erfüllt werden.

Anforderungsheader für serverseitige Verschlüsselung

Sie können die folgenden Anforderungsheader verwenden, um ein Objekt mit serverseitiger Verschlüsselung zu verschlüsseln. Die Optionen SSE und SSE-C schließen sich gegenseitig aus.

- **SSE**: Verwenden Sie den folgenden Header, wenn Sie das Objekt mit einem eindeutigen, von StorageGRID verwalteten Schlüssel verschlüsseln möchten.
  - `x-amz-server-side-encryption`

    Wenn die `x-amz-server-side-encryption` Header ist nicht in der PutObject-Anforderung enthalten, der rasterweite"Einstellung für die Verschlüsselung gespeicherter Objekte" wird aus der PutObject-Antwort weggelassen.

- **SSE-C**: Verwenden Sie alle drei Header, wenn Sie das Objekt mit einem eindeutigen Schlüssel verschlüsseln möchten, den Sie bereitstellen und verwalten.
  - `x-amz-server-side-encryption-customer-algorithm`: Angeben `AES256` .
  - `x-amz-server-side-encryption-customer-key`: Geben Sie Ihren Verschlüsselungsschlüssel für das neue Objekt an.
  - `x-amz-server-side-encryption-customer-key-MD5`: Geben Sie den MD5-Digest des Verschlüsselungsschlüssels des neuen Objekts an.

> Die von Ihnen bereitgestellten Verschlüsselungsschlüssel werden niemals gespeichert. Wenn Sie einen Verschlüsselungsschlüssel verlieren, verlieren Sie das entsprechende Objekt. Bevor Sie vom Kunden bereitgestellte Schlüssel zum Sichern von Objektdaten verwenden, lesen Sie die Überlegungen für"Verwendung serverseitiger Verschlüsselung" .

> Wenn ein Objekt mit SSE oder SSE-C verschlüsselt ist, werden alle Verschlüsselungseinstellungen auf Bucket- oder Grid-Ebene ignoriert.

Versionierung

Wenn die Versionierung für einen Bucket aktiviert ist, `versionId` wird automatisch für die Version des gespeicherten Objekts generiert. Das `versionId` wird auch in der Antwort zurückgegeben, indem der `x-amz-version-id` Antwortheader.

Wenn die Versionierung ausgesetzt ist, wird die Objektversion mit einem Nullwert gespeichert. `versionId` und wenn bereits eine Nullversion vorhanden ist, wird diese überschrieben.

Signaturberechnungen für den Autorisierungsheader

Bei Verwendung der `Authorization` Header zur Authentifizierung von Anfragen. StorageGRID unterscheidet sich in folgenden Punkten von AWS:

- StorageGRID erfordert nicht `host` Header, die in `CanonicalHeaders` .
- StorageGRID erfordert nicht `Content-Type` eingeschlossen sein in `CanonicalHeaders` .
- StorageGRID erfordert nicht `x-amz-*` Header, die in `CanonicalHeaders` .

> Als allgemeine Best Practice sollten Sie diese Header immer in `CanonicalHeaders` um sicherzustellen, dass sie überprüft werden. Wenn Sie diese Header jedoch ausschließen, gibt StorageGRID keinen Fehler zurück.

Weitere Einzelheiten finden Sie unter "Signaturberechnungen für den Autorisierungsheader: Übertragen der Nutzlast in einem einzigen Block (AWS-Signaturversion 4)" .

Ähnliche Informationen

- "Objekte mit ILM verwalten"
- "Amazon Simple Storage Service API-Referenz: PutObject"

RestoreObject

Sie können die S3 RestoreObject-Anforderung verwenden, um ein Objekt wiederherzustellen, das in einem Cloud-Speicherpool gespeichert ist.

Unterstützter Anfragetyp

StorageGRID unterstützt nur RestoreObject-Anfragen zum Wiederherstellen eines Objekts. Es unterstützt nicht die `SELECT` Art der Restaurierung. Wählen Sie Anfragen zurück `XNotImplemented` .

Versionierung

Geben Sie optional an `versionId` um eine bestimmte Version eines Objekts in einem versionierten Bucket wiederherzustellen. Wenn Sie nicht angeben `versionId` wird die aktuellste Version des Objekts wiederhergestellt

Verhalten von RestoreObject bei Cloud Storage Pool-Objekten

Wenn ein Objekt in einem"Cloud-Speicherpool" , eine RestoreObject-Anforderung weist basierend auf dem Status des Objekts das folgende Verhalten auf. Sehen"HeadObject" für weitere Details.

Wenn ein Objekt in einem Cloud-Speicherpool gespeichert ist und eine oder mehrere Kopien des Objekts auch im Grid vorhanden sind, ist es nicht erforderlich, das Objekt durch Ausgeben einer RestoreObject-Anforderung wiederherzustellen. Stattdessen kann die lokale Kopie direkt mithilfe einer GetObject-Anforderung abgerufen werden.

Zustand des Objekts	Verhalten von RestoreObject
Objekt in StorageGRID aufgenommen, aber noch nicht von ILM ausgewertet, oder Objekt befindet sich nicht in einem Cloud-Speicherpool	`403 Forbidden` , `InvalidObjectState`
Objekt im Cloud-Speicherpool, aber noch nicht in einen nicht abrufbaren Zustand übergegangen	`200 OK`Es werden keine Änderungen vorgenommen. **Hinweis**: Bevor ein Objekt in einen nicht abrufbaren Zustand überführt wurde, können Sie seine `expiry-date` .
Objekt in einen nicht abrufbaren Zustand überführt	`202 Accepted`Stellt eine abrufbare Kopie des Objekts für die im Anforderungstext angegebene Anzahl von Tagen im Cloud-Speicherpool wieder her. Am Ende dieses Zeitraums wird das Objekt in einen nicht abrufbaren Zustand zurückversetzt. Optional können Sie die `Tier` Anforderungselement, um zu bestimmen, wie lange es dauert, bis der Wiederherstellungsjob abgeschlossen ist(`Expedited` , `Standard` , oder `Bulk`). Wenn Sie nicht angeben `Tier` , Die `Standard` Ebene verwendet wird. **Wichtig**: Wenn ein Objekt in das S3 Glacier Deep Archive verschoben wurde oder der Cloud Storage Pool Azure Blob Storage verwendet, können Sie es nicht mit dem `Expedited` Stufe. Der folgende Fehler wird zurückgegeben `403 Forbidden` , `InvalidTier` : `Retrieval option is not supported by this storage class.`
Objekt wird gerade aus einem nicht abrufbaren Zustand wiederhergestellt	`409 Conflict` , `RestoreAlreadyInProgress`
Objekt vollständig im Cloud-Speicherpool wiederhergestellt	`200 OK` **Hinweis:** Wenn ein Objekt in einen abrufbaren Zustand zurückversetzt wurde, können Sie seine `expiry-date` durch erneutes Ausgeben der RestoreObject-Anforderung mit einem neuen Wert für `Days` . Das Wiederherstellungsdatum wird relativ zum Zeitpunkt der Anfrage aktualisiert.

SelectObjectContent

Sie können die S3 SelectObjectContent-Anforderung verwenden, um den Inhalt eines S3-Objekts basierend auf einer einfachen SQL-Anweisung zu filtern.

Weitere Informationen finden Sie unter "Amazon Simple Storage Service API-Referenz: SelectObjectContent" .

Bevor Sie beginnen

- Das Mandantenkonto verfügt über die Berechtigung „S3 Select“.
- Du hast `s3:GetObject` Berechtigung für das Objekt, das Sie abfragen möchten.
- Das abzufragende Objekt muss eines der folgenden Formate aufweisen:
  - **CSV**. Kann unverändert verwendet oder in GZIP- oder BZIP2-Archive komprimiert werden.
  - **Parkett**. Zusätzliche Anforderungen für Parquet-Objekte:
    - S3 Select unterstützt nur spaltenweise Komprimierung mit GZIP oder Snappy. S3 Select unterstützt keine Ganzobjektkomprimierung für Parquet-Objekte.
    - S3 Select unterstützt keine Parquet-Ausgabe. Sie müssen das Ausgabeformat als CSV oder JSON angeben.
    - Die maximale unkomprimierte Zeilengruppengröße beträgt 512 MB.
    - Sie müssen die im Schema des Objekts angegebenen Datentypen verwenden.
    - Sie können die logischen Typen INTERVAL, JSON, LIST, TIME oder UUID nicht verwenden.
- Ihr SQL-Ausdruck hat eine maximale Länge von 256 KB.
- Jeder Datensatz in der Eingabe oder den Ergebnissen hat eine maximale Länge von 1 MiB.

Beispiel für die CSV-Anforderungssyntax

```
POST /{Key+}?select&select-type=2 HTTP/1.1
Host: Bucket.s3.abc-company.com
x-amz-expected-bucket-owner: ExpectedBucketOwner
<?xml version="1.0" encoding="UTF-8"?>
<SelectObjectContentRequest xmlns="http://s3.amazonaws.com/doc/2006-03-
01/">
   <Expression>string</Expression>
   <ExpressionType>string</ExpressionType>
   <RequestProgress>
      <Enabled>boolean</Enabled>
   </RequestProgress>
   <InputSerialization>
      <CompressionType>GZIP</CompressionType>
      <CSV>
         <AllowQuotedRecordDelimiter>boolean</AllowQuotedRecordDelimiter>
         <Comments>#</Comments>
         <FieldDelimiter>\t</FieldDelimiter>
         <FileHeaderInfo>USE</FileHeaderInfo>
         <QuoteCharacter>'</QuoteCharacter>
         <QuoteEscapeCharacter>\\</QuoteEscapeCharacter>
         <RecordDelimiter>\n</RecordDelimiter>
      </CSV>
   </InputSerialization>
   <OutputSerialization>
      <CSV>
         <FieldDelimiter>string</FieldDelimiter>
         <QuoteCharacter>string</QuoteCharacter>
         <QuoteEscapeCharacter>string</QuoteEscapeCharacter>
         <QuoteFields>string</QuoteFields>
         <RecordDelimiter>string</RecordDelimiter>
      </CSV>
   </OutputSerialization>
   <ScanRange>
      <End>long</End>
      <Start>long</Start>
   </ScanRange>
</SelectObjectContentRequest>
```

Beispiel für die Parquet-Anforderungssyntax

```
POST /{Key+}?select&select-type=2 HTTP/1.1
Host: Bucket.s3.abc-company.com
x-amz-expected-bucket-owner: ExpectedBucketOwner
<?xml version="1.0" encoding="UTF-8"?>
<SelectObjectContentRequest xmlns=http://s3.amazonaws.com/doc/2006-03-01/>
   <Expression>string</Expression>
   <ExpressionType>string</ExpressionType>
   <RequestProgress>
      <Enabled>boolean</Enabled>
   </RequestProgress>
   <InputSerialization>
      <CompressionType>GZIP</CompressionType>
      <PARQUET>
      </PARQUET>
   </InputSerialization>
   <OutputSerialization>
      <CSV>
         <FieldDelimiter>string</FieldDelimiter>
         <QuoteCharacter>string</QuoteCharacter>
         <QuoteEscapeCharacter>string</QuoteEscapeCharacter>
         <QuoteFields>string</QuoteFields>
         <RecordDelimiter>string</RecordDelimiter>
      </CSV>
   </OutputSerialization>
   <ScanRange>
      <End>long</End>
      <Start>long</Start>
   </ScanRange>
</SelectObjectContentRequest>
```

SQL-Abfragebeispiel

Diese Abfrage ermittelt den Namen des Bundesstaates, die Bevölkerungszahlen von 2010, die geschätzten Bevölkerungszahlen von 2015 und die prozentuale Veränderung gegenüber den US-Volkszählungsdaten. Datensätze in der Datei, die keine Zustände sind, werden ignoriert.

```
SELECT STNAME, CENSUS2010POP, POPESTIMATE2015, CAST((POPESTIMATE2015 -
CENSUS2010POP) AS DECIMAL) / CENSUS2010POP * 100.0 FROM S3Object WHERE
NAME = STNAME
```

Die ersten Zeilen der abzufragenden Datei, `SUB-EST2020_ALL.csv` , sehen so aus:

```
SUMLEV,STATE,COUNTY,PLACE,COUSUB,CONCIT,PRIMGEO_FLAG,FUNCSTAT,NAME,STNAME,
CENSUS2010POP,
ESTIMATESBASE2010,POPESTIMATE2010,POPESTIMATE2011,POPESTIMATE2012,POPESTIM
ATE2013,POPESTIMATE2014,
POPESTIMATE2015,POPESTIMATE2016,POPESTIMATE2017,POPESTIMATE2018,POPESTIMAT
E2019,POPESTIMATE042020,
POPESTIMATE2020
040,01,000,00000,00000,00000,0,A,Alabama,Alabama,4779736,4780118,4785514,4
799642,4816632,4831586,
4843737,4854803,4866824,4877989,4891628,4907965,4920706,4921532
162,01,000,00124,00000,00000,0,A,Abbeville
city,Alabama,2688,2705,2699,2694,2645,2629,2610,2602,
2587,2578,2565,2555,2555,2553
162,01,000,00460,00000,00000,0,A,Adamsville
city,Alabama,4522,4487,4481,4474,4453,4430,4399,4371,
4335,4304,4285,4254,4224,4211
162,01,000,00484,00000,00000,0,A,Addison
town,Alabama,758,754,751,750,745,744,742,734,734,728,
725,723,719,717
```

AWS-CLI-Nutzungsbeispiel (CSV)

```
aws s3api select-object-content --endpoint-url https://10.224.7.44:10443
--no-verify-ssl  --bucket 619c0755-9e38-42e0-a614-05064f74126d --key SUB-
EST2020_ALL.csv --expression-type SQL --input-serialization '{"CSV":
{"FileHeaderInfo": "USE", "Comments": "#", "QuoteEscapeCharacter": "\"",
"RecordDelimiter": "\n", "FieldDelimiter": ",", "QuoteCharacter": "\"",
"AllowQuotedRecordDelimiter": false}, "CompressionType": "NONE"}' --output
-serialization '{"CSV": {"QuoteFields": "ASNEEDED",
"QuoteEscapeCharacter": "#", "RecordDelimiter": "\n", "FieldDelimiter":
",", "QuoteCharacter": "\""}}' --expression "SELECT STNAME, CENSUS2010POP,
POPESTIMATE2015, CAST((POPESTIMATE2015 - CENSUS2010POP) AS DECIMAL) /
CENSUS2010POP * 100.0 FROM S3Object WHERE NAME = STNAME" changes.csv
```

Die ersten paar Zeilen der Ausgabedatei, `changes.csv` , sehen so aus:

```
Alabama,4779736,4854803,1.570526070895965802295356898372629785 4
Alaska,710231,738430,3.970398363349389142405780654463125377 5
Arizona,6392017,6832810,6.895992297892824753125656580700583243 1
Arkansas,2915918,2979732,2.188470320495981025529524492801237894 9
California,37253956,38904296,4.429972483996062055798852610444914897 1
Colorado,5029196,5454328,8.453279609703022113276157859029554624 6
```

AWS-CLI-Nutzungsbeispiel (Parquet)

```
aws s3api select-object-content  -endpoint-url https://10.224.7.44:10443
--bucket 619c0755-9e38-42e0-a614-05064f74126d --key SUB-
EST2020_ALL.parquet --expression "SELECT STNAME, CENSUS2010POP,
POPESTIMATE2015, CAST((POPESTIMATE2015 - CENSUS2010POP) AS DECIMAL) /
CENSUS2010POP * 100.0 FROM S3Object WHERE NAME = STNAME" --expression-type
'SQL' --input-serialization '{"Parquet":{}}'  --output-serialization
'{"CSV": {}}' changes.csv
```

Die ersten Zeilen der Ausgabedatei changes.csv sehen folgendermaßen aus:

```
Alabama,4779736,4854803,1.5705260708959658022953568983726297854
Alaska,710231,738430,3.9703983633493891424057806544631253775
Arizona,6392017,6832810,6.8959922978928247531256565807005832431
Arkansas,2915918,2979732,2.1884703204959810255295244928012378949
California,37253956,38904296,4.4299724839960620557988526104449148971
Colorado,5029196,5454328,8.4532796097030221132761578590295546246
```

Vorgänge für mehrteilige Uploads

Vorgänge für mehrteilige Uploads

In diesem Abschnitt wird beschrieben, wie StorageGRID Vorgänge für mehrteilige Uploads unterstützt.

Für alle mehrteiligen Uploadvorgänge gelten die folgenden Bedingungen und Hinweise:

- Sie sollten nicht mehr als 1.000 gleichzeitige mehrteilige Uploads in einen einzelnen Bucket durchführen, da die Ergebnisse von ListMultipartUploads-Abfragen für diesen Bucket möglicherweise unvollständige Ergebnisse zurückgeben.
- StorageGRID erzwingt AWS-Größenbeschränkungen für mehrteilige Teile. S3-Clients müssen diese Richtlinien befolgen:
  - Jeder Teil eines mehrteiligen Uploads muss zwischen 5 MiB (5.242.880 Bytes) und 5 GiB (5.368.709.120 Bytes) groß sein.
  - Der letzte Teil kann kleiner als 5 MiB (5.242.880 Bytes) sein.
  - Generell sollten die Teilegrößen möglichst groß sein. Verwenden Sie beispielsweise Teilgrößen von 5 GiB für ein 100-GiB-Objekt. Da jedes Teil als einzigartiges Objekt betrachtet wird, reduziert die Verwendung großer Teilegrößen den StorageGRID Metadaten-Overhead.
  - Erwägen Sie für Objekte, die kleiner als 5 GiB sind, stattdessen die Verwendung eines nicht mehrteiligen Uploads.
- ILM wird für jeden Teil eines mehrteiligen Objekts ausgewertet, wenn es aufgenommen wird, und für das Objekt als Ganzes, wenn der mehrteilige Upload abgeschlossen ist, wenn die ILM-Regel die Balanced- oder Strict-Regel verwendet."Aufnahmeoption" . Sie sollten sich darüber im Klaren sein, welche Auswirkungen dies auf die Platzierung von Objekten und Teilen hat:

- Wenn sich ILM während eines laufenden S3-Multipart-Uploads ändert, erfüllen einige Teile des Objekts nach Abschluss des Multipart-Uploads möglicherweise nicht die aktuellen ILM-Anforderungen. Alle Teile, die nicht richtig platziert sind, werden zur erneuten ILM-Bewertung in die Warteschlange gestellt und später an die richtige Position verschoben.
- Bei der Auswertung von ILM für ein Teil filtert StorageGRID nach der Größe des Teils, nicht nach der Größe des Objekts. Dies bedeutet, dass Teile eines Objekts an Orten gespeichert werden können, die die ILM-Anforderungen für das gesamte Objekt nicht erfüllen. Wenn beispielsweise eine Regel angibt, dass alle Objekte mit 10 GB oder mehr bei DC1 gespeichert werden, während alle kleineren Objekte bei DC2 gespeichert werden, wird jeder 1-GB-Teil eines 10-teiligen mehrteiligen Uploads bei der Aufnahme bei DC2 gespeichert. Wenn ILM jedoch für das gesamte Objekt ausgewertet wird, werden alle Teile des Objekts nach DC1 verschoben.

- Alle mehrteiligen Upload-Vorgänge unterstützen StorageGRID"Konsistenzwerte" .
- Wenn ein Objekt per mehrteiligem Upload aufgenommen wird,"Schwellenwert für Objektsegmentierung (1 GiB)" wird nicht angewendet.
- Bei Bedarf können Sie"serverseitige Verschlüsselung" mit mehrteiligen Uploads. Um SSE (serverseitige Verschlüsselung mit StorageGRID-verwalteten Schlüsseln) zu verwenden, schließen Sie die `x-amz-server-side-encryption` Anforderungsheader nur in der CreateMultipartUpload-Anforderung. Um SSE-C (serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln) zu verwenden, geben Sie in der CreateMultipartUpload-Anforderung und in jeder nachfolgenden UploadPart-Anforderung dieselben drei Anforderungsheader für Verschlüsselungsschlüssel an.

Betrieb	Durchführung
AbortMultipartUpload	Implementiert mit dem gesamten Amazon S3 REST API-Verhalten. Änderungen vorbehalten.
CompleteMultipartUpload	Sehen"CompleteMultipartUpload"
CreateMultipartUpload (früher „Mehrteiligen Upload initiieren“ genannt)	Sehen"CreateMultipartUpload"
ListMultipartUploads	Sehen"ListMultipartUploads"
Teileliste	Implementiert mit dem gesamten Amazon S3 REST API-Verhalten. Änderungen vorbehalten.
UploadPart	Sehen"UploadPart"
UploadPartCopy	Sehen"UploadPartCopy"

CompleteMultipartUpload

Der Vorgang „CompleteMultipartUpload“ schließt einen mehrteiligen Upload eines Objekts ab, indem er die zuvor hochgeladenen Teile zusammenfügt.

StorageGRID unterstützt nicht aufeinanderfolgende Werte in aufsteigender Reihenfolge für die `partNumber` Anforderungsparameter mit CompleteMultipartUpload. Der Parameter kann mit einem beliebigen Wert beginnen.

Konflikte lösen

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.

Unterstützte Anforderungsheader

Die folgenden Anforderungsheader werden unterstützt:

- `x-amz-checksum-sha256`
- `x-amz-storage-class`

  Der `x-amz-storage-class` Der Header beeinflusst, wie viele Objektkopien StorageGRID erstellt, wenn die entsprechende ILM-Regel Folgendes angibt:"Option für doppeltes Commit oder ausgeglichene Aufnahme" .

- `STANDARD`

  (Standard) Gibt einen Dual-Commit-Aufnahmevorgang an, wenn die ILM-Regel die Option „Dual Commit“ verwendet oder wenn die Option „Balanced“ auf die Erstellung von Zwischenkopien zurückgreift.

- `REDUCED_REDUNDANCY`

  Gibt einen Single-Commit-Ingest-Vorgang an, wenn die ILM-Regel die Option „Dual Commit“ verwendet oder wenn die Option „Balanced“ auf die Erstellung von Zwischenkopien zurückgreift.

  Wenn Sie ein Objekt in einen Bucket mit aktivierter S3-Objektsperre aufnehmen, wird die `REDUCED_REDUNDANCY` Option wird ignoriert. Wenn Sie ein Objekt in einen Legacy-Compliant-Bucket aufnehmen, `REDUCED_REDUNDANCY` Option gibt einen Fehler zurück. StorageGRID führt immer eine Dual-Commit-Aufnahme durch, um sicherzustellen, dass die Compliance-Anforderungen erfüllt werden.

Wenn ein mehrteiliger Upload nicht innerhalb von 15 Tagen abgeschlossen wird, wird der Vorgang als inaktiv markiert und alle zugehörigen Daten werden aus dem System gelöscht.

Der `ETag` Der zurückgegebene Wert ist keine MD5-Summe der Daten, sondern folgt der Amazon S3 API-Implementierung des `ETag` Wert für mehrteilige Objekte.

Nicht unterstützte Anforderungsheader

Die folgenden Anforderungsheader werden nicht unterstützt:

- `x-amz-sdk-checksum-algorithm`
- `x-amz-trailer`

Versionierung

Dieser Vorgang schließt einen mehrteiligen Upload ab. Wenn die Versionierung für einen Bucket aktiviert ist, wird die Objektversion nach Abschluss des mehrteiligen Uploads erstellt.

Wenn die Versionierung für einen Bucket aktiviert ist, `versionId` wird automatisch für die Version des gespeicherten Objekts generiert. Das `versionId` wird auch in der Antwort zurückgegeben, indem der `x-amz-version-id` Antwortheader.

Wenn die Versionierung ausgesetzt ist, wird die Objektversion mit einem Nullwert gespeichert. `versionId` und wenn bereits eine Nullversion vorhanden ist, wird diese überschrieben.

> Wenn die Versionsverwaltung für einen Bucket aktiviert ist, wird beim Abschließen eines mehrteiligen Uploads immer eine neue Version erstellt, auch wenn gleichzeitig mehrteilige Uploads für denselben Objektschlüssel abgeschlossen wurden. Wenn die Versionsverwaltung für einen Bucket nicht aktiviert ist, ist es möglich, einen mehrteiligen Upload zu initiieren und dann zunächst einen weiteren mehrteiligen Upload mit demselben Objektschlüssel zu initiieren und abzuschließen. Bei Buckets ohne Versionsangabe hat der zuletzt abgeschlossene mehrteilige Upload Vorrang.

Fehlgeschlagene Replikation, Benachrichtigung oder Metadatenbenachrichtigung

Wenn der Bucket, in dem der mehrteilige Upload erfolgt, für einen Plattformdienst konfiguriert ist, ist der mehrteilige Upload auch dann erfolgreich, wenn die zugehörige Replikations- oder Benachrichtigungsaktion fehlschlägt.

Ein Mandant kann die fehlgeschlagene Replikation oder Benachrichtigung auslösen, indem er die Metadaten oder Tags des Objekts aktualisiert. Um unerwünschte Änderungen zu vermeiden, kann ein Mandant die vorhandenen Werte erneut übermitteln.

Weitere Informationen finden Sie unter "Fehlerbehebung bei Plattformdiensten" .

CreateMultipartUpload

Der Vorgang „CreateMultipartUpload“ (früher „Initiate Multipart Upload“) initiiert einen mehrteiligen Upload für ein Objekt und gibt eine Upload-ID zurück.

Der `x-amz-storage-class` Anforderungsheader wird unterstützt. Der übermittelte Wert für `x-amz-storage-class` beeinflusst, wie StorageGRID Objektdaten während der Aufnahme schützt, und nicht, wie viele persistente Kopien des Objekts im StorageGRID -System gespeichert werden (was durch ILM bestimmt wird).

Wenn die ILM-Regel, die einem aufgenommenen Objekt entspricht, die strikte"Aufnahmeoption" , Die `x-amz-storage-class` Header hat keine Wirkung.

Folgende Werte können verwendet werden für `x-amz-storage-class` :

- `STANDARD`(Standard)
  - **Dual Commit**: Wenn die ILM-Regel die Aufnahmeoption „Dual Commit“ angibt, wird, sobald ein Objekt aufgenommen wird, eine zweite Kopie dieses Objekts erstellt und an einen anderen Speicherknoten verteilt (Dual Commit). Bei der Auswertung des ILM ermittelt StorageGRID , ob diese ersten Zwischenkopien die Platzierungsanweisungen in der Regel erfüllen. Ist dies nicht der Fall, müssen möglicherweise neue Objektkopien an anderen Orten erstellt und die ersten Zwischenkopien gelöscht

werden.

- **Ausgeglichen**: Wenn die ILM-Regel die Option „Ausgeglichen“ angibt und StorageGRID nicht sofort alle in der Regel angegebenen Kopien erstellen kann, erstellt StorageGRID zwei Zwischenkopien auf verschiedenen Speicherknoten.

  Wenn StorageGRID alle in der ILM-Regel angegebenen Objektkopien sofort erstellen kann (synchrone Platzierung), `x-amz-storage-class` Header hat keine Wirkung.

- `REDUCED_REDUNDANCY`

  - **Dual Commit**: Wenn die ILM-Regel die Option „Dual Commit“ angibt, erstellt StorageGRID beim Einlesen des Objekts eine einzelne Zwischenkopie (Single Commit).
  - **Ausgeglichen**: Wenn die ILM-Regel die Option „Ausgeglichen“ angibt, erstellt StorageGRID nur dann eine einzelne Zwischenkopie, wenn das System nicht sofort alle in der Regel angegebenen Kopien erstellen kann. Wenn StorageGRID eine synchrone Platzierung durchführen kann, hat dieser Header keine Wirkung. Der `REDUCED_REDUNDANCY` Die Option wird am besten verwendet, wenn die ILM-Regel, die dem Objekt entspricht, eine einzelne replizierte Kopie erstellt. In diesem Fall mit `REDUCED_REDUNDANCY` vermeidet das unnötige Erstellen und Löschen einer zusätzlichen Objektkopie für jeden Aufnahmevorgang.

  Verwenden des `REDUCED_REDUNDANCY` Unter anderen Umständen wird diese Option nicht empfohlen. `REDUCED_REDUNDANCY` erhöht das Risiko eines Objektdatenverlusts während der Aufnahme. Beispielsweise können Daten verloren gehen, wenn die einzelne Kopie zunächst auf einem Speicherknoten gespeichert wird, der ausfällt, bevor die ILM-Auswertung erfolgen kann.

> Wenn für einen bestimmten Zeitraum nur eine Kopie vorhanden ist, besteht die Gefahr eines dauerhaften Datenverlusts. Wenn nur eine replizierte Kopie eines Objekts vorhanden ist, geht dieses Objekt verloren, wenn ein Speicherknoten ausfällt oder einen schwerwiegenden Fehler aufweist. Auch während Wartungsvorgängen wie Upgrades verlieren Sie vorübergehend den Zugriff auf das Objekt.

Festlegen `REDUCED_REDUNDANCY` wirkt sich nur darauf aus, wie viele Kopien erstellt werden, wenn ein Objekt zum ersten Mal aufgenommen wird. Es hat keinen Einfluss darauf, wie viele Kopien des Objekts erstellt werden, wenn das Objekt von den aktiven ILM-Richtlinien ausgewertet wird, und führt nicht dazu, dass Daten im StorageGRID System auf niedrigeren Redundanzebenen gespeichert werden.

> Wenn Sie ein Objekt in einen Bucket mit aktivierter S3-Objektsperre aufnehmen, wird die `REDUCED_REDUNDANCY` Option wird ignoriert. Wenn Sie ein Objekt in einen Legacy-Compliant-Bucket aufnehmen, `REDUCED_REDUNDANCY` Option gibt einen Fehler zurück. StorageGRID führt immer eine Dual-Commit-Aufnahme durch, um sicherzustellen, dass die Compliance-Anforderungen erfüllt werden.

Unterstützte Anforderungsheader

Die folgenden Anforderungsheader werden unterstützt:

- `Content-Type`
- `x-amz-checksum-algorithm`

  Derzeit ist nur der SHA256-Wert für `x-amz-checksum-algorithm` wird unterstützt.

- `x-amz-meta-`, gefolgt von einem Name-Wert-Paar mit benutzerdefinierten Metadaten

  Verwenden Sie beim Angeben des Name-Wert-Paares für benutzerdefinierte Metadaten dieses allgemeine Format:

  ```
  x-amz-meta-_name_: `value`
  ```

  Wenn Sie die Option **Benutzerdefinierte Erstellungszeit** als Referenzzeit für eine ILM-Regel verwenden möchten, müssen Sie `creation-time` als Name der Metadaten, die aufzeichnen, wann das Objekt erstellt wurde. Beispiel:

  ```
  x-amz-meta-creation-time: 1443399726
  ```

  Der Wert für `creation-time` wird seit dem 1. Januar 1970 in Sekunden ausgewertet.

  > Hinzufügen `creation-time` da benutzerdefinierte Metadaten nicht zulässig sind, wenn Sie ein Objekt zu einem Bucket hinzufügen, für den die Legacy-Compliance aktiviert ist. Es wird ein Fehler zurückgegeben.

- S3 Object Lock-Anforderungsheader:

  - `x-amz-object-lock-mode`
  - `x-amz-object-lock-retain-until-date`
  - `x-amz-object-lock-legal-hold`

    Wenn eine Anfrage ohne diese Header gestellt wird, werden die Bucket-Standardaufbewahrungseinstellungen verwendet, um das Aufbewahrungsdatum der Objektversion zu berechnen.

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

- SSE-Anforderungsheader:

  - `x-amz-server-side-encryption`
  - `x-amz-server-side-encryption-customer-key-MD5`
  - `x-amz-server-side-encryption-customer-key`
  - `x-amz-server-side-encryption-customer-algorithm`

    Anforderungsheader für serverseitige Verschlüsselung

  > Informationen zur Verarbeitung von UTF-8-Zeichen durch StorageGRID finden Sie unter"PutObject" .

Anforderungsheader für serverseitige Verschlüsselung

Sie können die folgenden Anforderungsheader verwenden, um ein mehrteiliges Objekt mit serverseitiger

Verschlüsselung zu verschlüsseln. Die Optionen SSE und SSE-C schließen sich gegenseitig aus.

- **SSE**: Verwenden Sie den folgenden Header in der CreateMultipartUpload-Anforderung, wenn Sie das Objekt mit einem eindeutigen, von StorageGRID verwalteten Schlüssel verschlüsseln möchten. Geben Sie diesen Header in keiner der UploadPart-Anfragen an.
  - `x-amz-server-side-encryption`
- **SSE-C**: Verwenden Sie alle drei dieser Header in der CreateMultipartUpload-Anfrage (und in jeder nachfolgenden UploadPart-Anfrage), wenn Sie das Objekt mit einem eindeutigen Schlüssel verschlüsseln möchten, den Sie bereitstellen und verwalten.
  - `x-amz-server-side-encryption-customer-algorithm`: Angeben `AES256` .
  - `x-amz-server-side-encryption-customer-key`: Geben Sie Ihren Verschlüsselungsschlüssel für das neue Objekt an.
  - `x-amz-server-side-encryption-customer-key-MD5`: Geben Sie den MD5-Digest des Verschlüsselungsschlüssels des neuen Objekts an.

> ⚠ Die von Ihnen bereitgestellten Verschlüsselungsschlüssel werden niemals gespeichert. Wenn Sie einen Verschlüsselungsschlüssel verlieren, verlieren Sie das entsprechende Objekt. Bevor Sie vom Kunden bereitgestellte Schlüssel zum Sichern von Objektdaten verwenden, lesen Sie die Überlegungen für"Verwendung serverseitiger Verschlüsselung" .

Nicht unterstützte Anforderungsheader

Der folgende Anforderungsheader wird nicht unterstützt:

- `x-amz-website-redirect-location`

  Der `x-amz-website-redirect-location` Header-Returns `XNotImplemented` .

Versionierung

Der mehrteilige Upload besteht aus separaten Vorgängen zum Starten des Uploads, Auflisten der Uploads, Hochladen von Teilen, Zusammenstellen der hochgeladenen Teile und Abschließen des Uploads. Objekte werden erstellt (und gegebenenfalls versioniert), wenn der Vorgang CompleteMultipartUpload ausgeführt wird.

ListMultipartUploads

Der Vorgang „ListMultipartUploads“ listet laufende mehrteilige Uploads für einen Bucket auf.

Die folgenden Anforderungsparameter werden unterstützt:

- `encoding-type`
- `key-marker`
- `max-uploads`
- `prefix`
- `upload-id-marker`
- `Host`

- `Date`
- `Authorization`

Versionierung

Der mehrteilige Upload besteht aus separaten Vorgängen zum Starten des Uploads, Auflisten der Uploads, Hochladen von Teilen, Zusammenstellen der hochgeladenen Teile und Abschließen des Uploads. Objekte werden erstellt (und gegebenenfalls versioniert), wenn der Vorgang CompleteMultipartUpload ausgeführt wird.

UploadPart

Der Vorgang „UploadPart“ lädt einen Teil in einem mehrteiligen Upload für ein Objekt hoch.

Unterstützte Anforderungsheader

Die folgenden Anforderungsheader werden unterstützt:

- `x-amz-checksum-sha256`
- `Content-Length`
- `Content-MD5`

Anforderungsheader für serverseitige Verschlüsselung

Wenn Sie für die CreateMultipartUpload-Anforderung die SSE-C-Verschlüsselung angegeben haben, müssen Sie in jede UploadPart-Anforderung auch die folgenden Anforderungsheader einfügen:

- `x-amz-server-side-encryption-customer-algorithm`: Angeben `AES256` .
- `x-amz-server-side-encryption-customer-key`: Geben Sie denselben Verschlüsselungsschlüssel an, den Sie in der CreateMultipartUpload-Anforderung angegeben haben.
- `x-amz-server-side-encryption-customer-key-MD5`: Geben Sie denselben MD5-Digest an, den Sie in der CreateMultipartUpload-Anforderung angegeben haben.

> Die von Ihnen bereitgestellten Verschlüsselungsschlüssel werden niemals gespeichert. Wenn Sie einen Verschlüsselungsschlüssel verlieren, verlieren Sie das entsprechende Objekt. Bevor Sie vom Kunden bereitgestellte Schlüssel zum Sichern von Objektdaten verwenden, lesen Sie die Hinweise in"Verwenden Sie serverseitige Verschlüsselung" .

Wenn Sie während der CreateMultipartUpload-Anforderung eine SHA-256-Prüfsumme angegeben haben, müssen Sie in jede UploadPart-Anforderung auch den folgenden Anforderungsheader einfügen:

- `x-amz-checksum-sha256`: Geben Sie die SHA-256-Prüfsumme für diesen Teil an.

Nicht unterstützte Anforderungsheader

Die folgenden Anforderungsheader werden nicht unterstützt:

- `x-amz-sdk-checksum-algorithm`
- `x-amz-trailer`

Versionierung

Der mehrteilige Upload besteht aus separaten Vorgängen zum Starten des Uploads, Auflisten der Uploads, Hochladen von Teilen, Zusammenstellen der hochgeladenen Teile und Abschließen des Uploads. Objekte werden erstellt (und gegebenenfalls versioniert), wenn der Vorgang CompleteMultipartUpload ausgeführt wird.

UploadPartCopy

Der Vorgang „UploadPartCopy“ lädt einen Teil eines Objekts hoch, indem Daten aus einem vorhandenen Objekt als Datenquelle kopiert werden.

Der Vorgang „UploadPartCopy“ wird mit dem gesamten Amazon S3 REST-API-Verhalten implementiert. Änderungen vorbehalten.

Diese Anfrage liest und schreibt die Objektdaten, die in `x-amz-copy-source-range` innerhalb des StorageGRID -Systems.

Die folgenden Anforderungsheader werden unterstützt:

- `x-amz-copy-source-if-match`
- `x-amz-copy-source-if-none-match`
- `x-amz-copy-source-if-unmodified-since`
- `x-amz-copy-source-if-modified-since`

Anforderungsheader für serverseitige Verschlüsselung

Wenn Sie für die CreateMultipartUpload-Anforderung eine SSE-C-Verschlüsselung angegeben haben, müssen Sie in jede UploadPartCopy-Anforderung auch die folgenden Anforderungsheader einfügen:

- `x-amz-server-side-encryption-customer-algorithm`: Angeben `AES256` .
- `x-amz-server-side-encryption-customer-key`: Geben Sie denselben Verschlüsselungsschlüssel an, den Sie in der CreateMultipartUpload-Anforderung angegeben haben.
- `x-amz-server-side-encryption-customer-key-MD5`: Geben Sie denselben MD5-Digest an, den Sie in der CreateMultipartUpload-Anforderung angegeben haben.

Wenn das Quellobjekt mit einem vom Kunden bereitgestellten Schlüssel (SSE-C) verschlüsselt ist, müssen Sie die folgenden drei Header in die UploadPartCopy-Anforderung aufnehmen, damit das Objekt entschlüsselt und dann kopiert werden kann:

- `x-amz-copy-source-server-side-encryption-customer-algorithm`: Angeben `AES256` .
- `x-amz-copy-source-server-side-encryption-customer-key`: Geben Sie den Verschlüsselungsschlüssel an, den Sie beim Erstellen des Quellobjekts angegeben haben.
- `x-amz-copy-source-server-side-encryption-customer-key-MD5`: Geben Sie den MD5-Digest an, den Sie beim Erstellen des Quellobjekts angegeben haben.

> Die von Ihnen bereitgestellten Verschlüsselungsschlüssel werden niemals gespeichert. Wenn Sie einen Verschlüsselungsschlüssel verlieren, verlieren Sie das entsprechende Objekt. Bevor Sie vom Kunden bereitgestellte Schlüssel zum Sichern von Objektdaten verwenden, lesen Sie die Hinweise in"Verwenden Sie serverseitige Verschlüsselung" .

Versionierung

Der mehrteilige Upload besteht aus separaten Vorgängen zum Starten des Uploads, Auflisten der Uploads, Hochladen von Teilen, Zusammenstellen der hochgeladenen Teile und Abschließen des Uploads. Objekte werden erstellt (und gegebenenfalls versioniert), wenn der Vorgang CompleteMultipartUpload ausgeführt wird.

Fehlerantworten

Das StorageGRID -System unterstützt alle zutreffenden Standardfehlerantworten der S3 REST-API. Darüber hinaus fügt die StorageGRID -Implementierung mehrere benutzerdefinierte Antworten hinzu.

Unterstützte S3-API-Fehlercodes

Name	HTTP-Status
Zugriff verweigert	403 Verboten
BadDigest	400 Ungültige Anfrage
BucketExistsAlready	409 Konflikt
EimerNichtLeer	409 Konflikt
Unvollständiger Körper	400 Ungültige Anfrage
Interner Fehler	500 Interner Serverfehler
Ungültige Zugriffsschlüssel-ID	403 Verboten
Ungültiges Argument	400 Ungültige Anfrage
Ungültiger BucketName	400 Ungültige Anfrage
Ungültiger BucketState	409 Konflikt
InvalidDigest	400 Ungültige Anfrage
Fehler „Ungültiger Verschlüsselungsalgorithmus“	400 Ungültige Anfrage
UngültigesTeil	400 Ungültige Anfrage
UngültigeTeilebestellung	400 Ungültige Anfrage
Ungültiger Bereich	416 Angeforderter Bereich nicht erfüllbar
Ungültige Anfrage	400 Ungültige Anfrage

Name	HTTP-Status
Ungültige Speicherklasse	400 Ungültige Anfrage
Ungültiges Tag	400 Ungültige Anfrage
Ungültige URI	400 Ungültige Anfrage
Schlüssel zu lang	400 Ungültige Anfrage
MalformedXML	400 Ungültige Anfrage
Metadaten zu groß	400 Ungültige Anfrage
MethodeNichtZulässig	405 Methode nicht zulässig
MissingContentLength	411 Erforderliche Länge
MissingRequestBodyError	400 Ungültige Anfrage
MissingSecurityHeader	400 Ungültige Anfrage
KeinSuchBucket	404 Nicht gefunden
NoSuchKey	404 Nicht gefunden
NoSuchUpload	404 Nicht gefunden
Nicht implementiert	501 Nicht implementiert
NoSuchBucketPolicy	404 Nicht gefunden
ObjectLockConfigurationNotFoundError	404 Nicht gefunden
Vorbedingung fehlgeschlagen	412 Vorbedingung fehlgeschlagen
RequestTimeTooSkewed	403 Verboten
Dienst nicht verfügbar	503 Dienst nicht verfügbar
Signatur stimmt nicht überein	403 Verboten
Zu viele Eimer	400 Ungültige Anfrage
Benutzerschlüssel muss angegeben werden	400 Ungültige Anfrage

Benutzerdefinierte StorageGRID -Fehlercodes

Name	Beschreibung	HTTP-Status
XBucketLifecycleNotAllowed	Die Bucket-Lebenszykluskonfiguration ist in einem älteren konformen Bucket nicht zulässig	400 Ungültige Anfrage
XBucketPolicyParseException	Das Parsen der empfangenen Bucket-Richtlinien-JSON ist fehlgeschlagen.	400 Ungültige Anfrage
XComplianceConflict	Vorgang aufgrund veralteter Compliance-Einstellungen abgelehnt.	403 Verboten
XComplianceReduzierteRedundanzVerboten	Reduzierte Redundanz ist im Legacy-Compliant-Bucket nicht zulässig	400 Ungültige Anfrage
XMaxBucketPolicyLengthExceeded	Ihre Richtlinie überschreitet die maximal zulässige Bucket-Richtlinienlänge.	400 Ungültige Anfrage
XMissingInternalRequestHeader	Es fehlt ein Header einer internen Anfrage.	400 Ungültige Anfrage
XNoSuchBucketCompliance	Für den angegebenen Bucket ist die Legacy-Compliance nicht aktiviert.	404 Nicht gefunden
XNichtAkzeptabel	Die Anfrage enthält einen oder mehrere Accept-Header, die nicht erfüllt werden konnten.	406 Nicht akzeptabel
XNotImplemented	Die von Ihnen angegebene Anfrage impliziert eine Funktionalität, die nicht implementiert ist.	501 Nicht implementiert

Benutzerdefinierte StorageGRID -Vorgänge

Benutzerdefinierte StorageGRID -Vorgänge

Das StorageGRID -System unterstützt benutzerdefinierte Vorgänge, die der S3 REST-API hinzugefügt werden.

In der folgenden Tabelle sind die von StorageGRID unterstützten benutzerdefinierten Vorgänge aufgeführt.

Betrieb	Beschreibung
"GET Bucket-Konsistenz"	Gibt die Konsistenz zurück, die auf einen bestimmten Bucket angewendet wird.

Betrieb	Beschreibung
"PUT Bucket-Konsistenz"	Legt die Konsistenz fest, die auf einen bestimmten Bucket angewendet wird.
"GET Bucket – Letzte Zugriffszeit"	Gibt zurück, ob Aktualisierungen der letzten Zugriffszeit für einen bestimmten Bucket aktiviert oder deaktiviert sind.
"PUT Bucket: Letzte Zugriffszeit"	Ermöglicht Ihnen, Aktualisierungen der letzten Zugriffszeit für einen bestimmten Bucket zu aktivieren oder zu deaktivieren.
"Konfiguration der Benachrichtigung über DELETE-Bucket-Metadaten"	Löscht die XML-Metadatenbenachrichtigungskonfiguration, die einem bestimmten Bucket zugeordnet ist.
"GET Bucket-Metadaten-Benachrichtigungskonfiguration"	Gibt die XML-Konfigurationsdatei für Metadatenbenachrichtigungen zurück, die einem bestimmten Bucket zugeordnet ist.
"Konfiguration der Benachrichtigung über PUT-Bucket-Metadaten"	Konfiguriert den Metadaten-Benachrichtigungsdienst für einen Bucket.
"GET-Speichernutzung"	Gibt die Gesamtspeichermenge an, die von einem Konto und jedem mit dem Konto verknüpften Bucket verwendet wird.
"Veraltet: CreateBucket mit Compliance-Einstellungen"	Veraltet und nicht unterstützt: Sie können keine neuen Buckets mehr erstellen, wenn Compliance aktiviert ist.
"Veraltet: GET Bucket-Konformität"	Veraltet, aber unterstützt: Gibt die aktuell gültigen Compliance-Einstellungen für einen vorhandenen Legacy-Compliant-Bucket zurück.
"Veraltet: PUT-Bucket-Konformität"	Veraltet, aber unterstützt: Ermöglicht Ihnen, die Compliance-Einstellungen für einen vorhandenen Legacy-Compliant-Bucket zu ändern.

GET Bucket-Konsistenz

Mit der Anforderung „GET Bucket Consistency“ können Sie die Konsistenz ermitteln, die auf einen bestimmten Bucket angewendet wird.

Die Standardkonsistenz ist so eingestellt, dass für neu erstellte Objekte das Lesen nach dem Schreiben gewährleistet ist.

Sie müssen über die Berechtigung s3:GetBucketConsistency verfügen oder Root-Kontoinhaber sein, um diesen Vorgang abzuschließen.

Anforderungsbeispiel

```
GET /bucket?x-ntap-sg-consistency HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

Antwort

In der Antwort-XML `<Consistency>` gibt einen der folgenden Werte zurück:

Konsistenz	Beschreibung
alle	Alle Knoten empfangen die Daten sofort, andernfalls schlägt die Anforderung fehl.
stark-global	Garantiert Lese- und Schreibkonsistenz für alle Clientanforderungen auf allen Sites.
starke Site	Garantiert die Lese- und Schreibkonsistenz für alle Clientanforderungen innerhalb einer Site.
Lesen nach neuem Schreiben	(Standard) Bietet Read-After-Write-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.

Antwortbeispiel

```
HTTP/1.1 200 OK
Date: Fri, 18 Sep 2020 01:02:18 GMT
Connection: CLOSE
Server: StorageGRID/11.5.0
x-amz-request-id: 12345
Content-Length: 127
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<Consistency xmlns="http://s3.storagegrid.com/doc/2015-02-01/">read-after-
new-write</Consistency>
```

Ähnliche Informationen

"Konsistenzwerte"

PUT Bucket-Konsistenz

Mit der PUT-Bucket-Konsistenzanforderung können Sie die Konsistenz angeben, die auf Vorgänge angewendet werden soll, die an einem Bucket ausgeführt werden.

Die Standardkonsistenz ist so eingestellt, dass für neu erstellte Objekte das Lesen nach dem Schreiben gewährleistet ist.

Bevor Sie beginnen

Sie müssen über die Berechtigung „s3:PutBucketConsistency“ verfügen oder Root-Kontoinhaber sein, um diesen Vorgang abzuschließen.

Anfrage

Der `x-ntap-sg-consistency` Der Parameter muss einen der folgenden Werte enthalten:

Konsistenz	Beschreibung
alle	Alle Knoten empfangen die Daten sofort, andernfalls schlägt die Anforderung fehl.
stark-global	Garantiert Lese- und Schreibkonsistenz für alle Clientanforderungen auf allen Sites.
starke Site	Garantiert die Lese- und Schreibkonsistenz für alle Clientanforderungen innerhalb einer Site.
Lesen nach neuem Schreiben	(Standard) Bietet Read-After-Write-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.

Hinweis: Im Allgemeinen sollten Sie die Konsistenz „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.

Anforderungsbeispiel

```
PUT /bucket?x-ntap-sg-consistency=strong-global HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

Ähnliche Informationen

"Konsistenzwerte"

GET Bucket – Letzte Zugriffszeit

Mit der Anforderung „GET Bucket last access time“ können Sie feststellen, ob Aktualisierungen der letzten Zugriffszeit für einzelne Buckets aktiviert oder deaktiviert sind.

Sie müssen über die Berechtigung s3:GetBucketLastAccessTime verfügen oder Root-Kontobenutzer sein, um diesen Vorgang abzuschließen.

Anforderungsbeispiel

```
GET /bucket?x-ntap-sg-lastaccesstime HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

Antwortbeispiel

Dieses Beispiel zeigt, dass Aktualisierungen der letzten Zugriffszeit für den Bucket aktiviert sind.

```
HTTP/1.1 200 OK
Date: Sat, 29 Nov 2015 01:02:18 GMT
Connection: CLOSE
Server: StorageGRID/10.3.0
x-amz-request-id: 12345
Content-Length: 127
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<LastAccessTime xmlns="http://s3.storagegrid.com/doc/2015-02-01/">enabled
</LastAccessTime>
```

PUT Bucket: Letzte Zugriffszeit

Mit der Anforderung „PUT Bucket Last Access Time“ können Sie Aktualisierungen der letzten Zugriffszeit für einzelne Buckets aktivieren oder deaktivieren. Das Deaktivieren

der Aktualisierung der letzten Zugriffszeit verbessert die Leistung und ist die Standardeinstellung für alle Buckets, die mit Version 10.3.0 oder höher erstellt wurden.

Sie müssen über die Berechtigung s3:PutBucketLastAccessTime für einen Bucket verfügen oder Konto-Root sein, um diesen Vorgang abzuschließen.

> Ab StorageGRID Version 10.3 sind Aktualisierungen der letzten Zugriffszeit für alle neuen Buckets standardmäßig deaktiviert. Wenn Sie Buckets haben, die mit einer früheren Version von StorageGRID erstellt wurden, und Sie das neue Standardverhalten übernehmen möchten, müssen Sie die Aktualisierung der letzten Zugriffszeit für jeden dieser früheren Buckets explizit deaktivieren. Sie können Aktualisierungen der letzten Zugriffszeit mithilfe der Anforderung „PUT Bucket-Letztzugriffszeit“ oder auf der Detailseite für einen Bucket im Mandanten-Manager aktivieren oder deaktivieren. Sehen "Aktivieren oder Deaktivieren der Aktualisierung der letzten Zugriffszeit" .

Wenn die Aktualisierung der letzten Zugriffszeit für einen Bucket deaktiviert ist, wird das folgende Verhalten auf Vorgänge im Bucket angewendet:

- GetObject-, GetObjectAcl-, GetObjectTagging- und HeadObject-Anfragen aktualisieren die letzte Zugriffszeit nicht. Das Objekt wird nicht zu Warteschlangen für die Auswertung des Information Lifecycle Management (ILM) hinzugefügt.
- CopyObject- und PutObjectTagging-Anfragen, die nur die Metadaten aktualisieren, aktualisieren auch die letzte Zugriffszeit. Das Objekt wird zur ILM-Auswertung zu Warteschlangen hinzugefügt.
- Wenn Aktualisierungen der letzten Zugriffszeit für den Quell-Bucket deaktiviert sind, aktualisieren CopyObject-Anfragen die letzte Zugriffszeit für den Quell-Bucket nicht. Das kopierte Objekt wird nicht zu den Warteschlangen für die ILM-Auswertung für den Quell-Bucket hinzugefügt. Für das Ziel aktualisieren CopyObject-Anfragen jedoch immer die letzte Zugriffszeit. Die Kopie des Objekts wird zur ILM-Auswertung zu Warteschlangen hinzugefügt.
- CompleteMultipartUpload-Anfragen aktualisieren die letzte Zugriffszeit. Das fertige Objekt wird zur ILM-Auswertung in die Warteschlangen aufgenommen.

Anforderungsbeispiele

Dieses Beispiel aktiviert die letzte Zugriffszeit für einen Bucket.

```
PUT /bucket?x-ntap-sg-lastaccesstime=enabled HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

Dieses Beispiel deaktiviert die letzte Zugriffszeit für einen Bucket.

```
PUT /bucket?x-ntap-sg-lastaccesstime=disabled HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

Konfiguration der Benachrichtigung über DELETE-Bucket-Metadaten

Mit der Konfigurationsanforderung „DELETE Bucket-Metadaten-Benachrichtigung“ können Sie den Suchintegrationsdienst für einzelne Buckets deaktivieren, indem Sie die Konfigurations-XML löschen.

Sie müssen über die Berechtigung s3:DeleteBucketMetadataNotification für einen Bucket verfügen oder Konto-Root sein, um diesen Vorgang abzuschließen.

Anforderungsbeispiel

Dieses Beispiel zeigt das Deaktivieren des Suchintegrationsdienstes für einen Bucket.

```
DELETE /test1?x-ntap-sg-metadata-notification HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

GET Bucket-Metadaten-Benachrichtigungskonfiguration

Mit der Konfigurationsanforderung „GET Bucket-Metadatenbenachrichtigung“ können Sie die Konfigurations-XML abrufen, die zum Konfigurieren der Suchintegration für einzelne Buckets verwendet wird.

Sie müssen über die Berechtigung s3:GetBucketMetadataNotification verfügen oder Root-Kontoinhaber sein, um diesen Vorgang abzuschließen.

Anforderungsbeispiel

Diese Anfrage ruft die Metadaten-Benachrichtigungskonfiguration für den Bucket mit dem Namen ab. `bucket` .

```
GET /bucket?x-ntap-sg-metadata-notification HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

Antwort

Der Antworttext enthält die Metadatenbenachrichtigungskonfiguration für den Bucket. Mit der Konfiguration der Metadatenbenachrichtigung können Sie festlegen, wie der Bucket für die Suchintegration konfiguriert wird. Das heißt, Sie können feststellen, welche Objekte indiziert werden und an welche Endpunkte ihre Objektmetadaten gesendet werden.

```
<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>rule-status</Status>
        <Prefix>key-prefix</Prefix>
        <Destination>
           <Urn>arn:aws:es:_region:account-
ID_:domain/_mydomain/myindex/mytype_</Urn>
        </Destination>
    </Rule>
    <Rule>
        <ID>Rule-2</ID>
         ...
    </Rule>
     ...
</MetadataNotificationConfiguration>
```

Jede Metadatenbenachrichtigungskonfiguration umfasst eine oder mehrere Regeln. Jede Regel gibt die Objekte an, auf die sie angewendet wird, und das Ziel, an das StorageGRID Objektmetadaten senden soll. Ziele müssen mithilfe der URN eines StorageGRID Endpunkts angegeben werden.

Name	Beschreibung	Erforderlich
Metadatenbenachrichtigungskonfiguration	Container-Tag für Regeln, die zum Angeben der Objekte und des Ziels für Metadatenbenachrichtigungen verwendet werden. Enthält ein oder mehrere Regelelemente.	Ja
Regel	Container-Tag für eine Regel, die die Objekte identifiziert, deren Metadaten einem angegebenen Index hinzugefügt werden sollen. Regeln mit überlappenden Präfixen werden abgelehnt. Im MetadataNotificationConfiguration-Element enthalten.	Ja
AUSWEIS	Eindeutige Kennung für die Regel. Im Regelelement enthalten.	Nein
Status	Der Status kann „Aktiviert“ oder „Deaktiviert“ sein. Für deaktivierte Regeln werden keine Maßnahmen ergriffen. Im Regelelement enthalten.	Ja

Name	Beschreibung	Erforderlich
Präfix	Objekte, die dem Präfix entsprechen, sind von der Regel betroffen und ihre Metadaten werden an das angegebene Ziel gesendet. Um alle Objekte abzugleichen, geben Sie ein leeres Präfix an. Im Regelelement enthalten.	Ja
Ziel	Container-Tag für das Ziel einer Regel. Im Regelelement enthalten.	Ja
Urne	URN des Ziels, an das die Objektmetadaten gesendet werden. Muss die URN eines StorageGRID -Endpunkts mit den folgenden Eigenschaften sein: • `es`muss das dritte Element sein. • Die URN muss mit dem Index und Typ enden, in dem die Metadaten gespeichert sind, in der Form `domain-name/myindex/mytype` . Endpunkte werden mithilfe des Tenant Managers oder der Tenant Management API konfiguriert. Sie haben folgende Form: • `arn:aws:es:_region:account-ID_:domain/mydomain/myindex/mytype` • `urn:mysite:es:::mydomain/myindex/mytype` Der Endpunkt muss konfiguriert werden, bevor die Konfigurations-XML übermittelt wird, andernfalls schlägt die Konfiguration mit einem 404-Fehler fehl. Die Urne ist im Zielelement enthalten.	Ja

Antwortbeispiel

Das XML, das zwischen den `<MetadataNotificationConfiguration></MetadataNotificationConfiguration>` Tags zeigen, wie die Integration mit einem Suchintegrationsendpunkt für den Bucket konfiguriert ist. In diesem Beispiel werden Objektmetadaten an einen Elasticsearch-Index namens gesendet. `current` und geben Sie den Namen ein `2017` das in einer AWS-Domäne namens gehostet wird `records` .

```
HTTP/1.1 200 OK
Date: Thu, 20 Jul 2017 18:24:05 GMT
Connection: KEEP-ALIVE
Server: StorageGRID/11.0.0
x-amz-request-id: 3832973499
Content-Length: 264
Content-Type: application/xml

<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>Enabled</Status>
        <Prefix>2017</Prefix>
        <Destination>
           <Urn>arn:aws:es:us-east-
1:3333333:domain/records/current/2017</Urn>
        </Destination>
    </Rule>
</MetadataNotificationConfiguration>
```

Ähnliche Informationen

"Verwenden eines Mandantenkontos"

Konfiguration der Benachrichtigung über PUT-Bucket-Metadaten

Mit der Konfigurationsanforderung für die Benachrichtigung über PUT-Bucket-Metadaten können Sie den Suchintegrationsdienst für einzelne Buckets aktivieren. Die XML-Konfigurations-XML für die Metadatenbenachrichtigung, die Sie im Anforderungstext angeben, gibt die Objekte an, deren Metadaten an den Zielsuchindex gesendet werden.

Sie müssen über die Berechtigung s3:PutBucketMetadataNotification für einen Bucket verfügen oder Konto-Root sein, um diesen Vorgang abzuschließen.

Anfrage

Die Anfrage muss die Metadatenbenachrichtigungskonfiguration im Anfragetext enthalten. Jede Metadatenbenachrichtigungskonfiguration umfasst eine oder mehrere Regeln. Jede Regel gibt die Objekte an, auf die sie angewendet wird, und das Ziel, an das StorageGRID Objektmetadaten senden soll.

Objekte können nach dem Präfix des Objektnamens gefiltert werden. Beispielsweise könnten Sie Metadaten für Objekte mit dem Präfix `/images` zu einem Ziel und Objekte mit dem Präfix `/videos` zu einem anderen.

Konfigurationen mit überlappenden Präfixen sind ungültig und werden bei der Übermittlung abgelehnt. Beispielsweise eine Konfiguration, die eine Regel für Objekte mit dem Präfix `test` und eine zweite Regel für Objekte mit dem Präfix `test2` wäre nicht erlaubt.

Ziele müssen mithilfe der URN eines StorageGRID Endpunkts angegeben werden. Der Endpunkt muss vorhanden sein, wenn die Konfiguration der Metadatenbenachrichtigung übermittelt wird, sonst schlägt die

Anfrage fehl, da `400 Bad Request` Die Fehlermeldung lautet: `Unable to save the metadata notification (search) policy. The specified endpoint URN does not exist: URN.`

```
<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>rule-status</Status>
        <Prefix>key-prefix</Prefix>
        <Destination>
           <Urn>arn:aws:es:region:account-
ID:domain/mydomain/myindex/mytype</Urn>
        </Destination>
    </Rule>
    <Rule>
        <ID>Rule-2</ID>
         ...
    </Rule>
     ...
</MetadataNotificationConfiguration>
```

Die Tabelle beschreibt die Elemente in der XML-Konfiguration der Metadatenbenachrichtigung.

Name	Beschreibung	Erforderlich
Metadatenbenachrichtigungskonfiguration	Container-Tag für Regeln, die zum Angeben der Objekte und des Ziels für Metadatenbenachrichtigungen verwendet werden. Enthält ein oder mehrere Regelelemente.	Ja
Regel	Container-Tag für eine Regel, die die Objekte identifiziert, deren Metadaten einem angegebenen Index hinzugefügt werden sollen. Regeln mit überlappenden Präfixen werden abgelehnt. Im MetadataNotificationConfiguration-Element enthalten.	Ja
AUSWEIS	Eindeutige Kennung für die Regel. Im Regelelement enthalten.	Nein

Name	Beschreibung	Erforderlich
Status	Der Status kann „Aktiviert“ oder „Deaktiviert“ sein. Für deaktivierte Regeln werden keine Maßnahmen ergriffen. Im Regelelement enthalten.	Ja
Präfix	Objekte, die dem Präfix entsprechen, sind von der Regel betroffen und ihre Metadaten werden an das angegebene Ziel gesendet. Um alle Objekte abzugleichen, geben Sie ein leeres Präfix an. Im Regelelement enthalten.	Ja
Ziel	Container-Tag für das Ziel einer Regel. Im Regelelement enthalten.	Ja
Urne	URN des Ziels, an das die Objektmetadaten gesendet werden. Muss die URN eines StorageGRID -Endpunkts mit den folgenden Eigenschaften sein: • `es`muss das dritte Element sein. • Die URN muss mit dem Index und Typ enden, in dem die Metadaten gespeichert sind, in der Form `domain-name/myindex/mytype` . Endpunkte werden mithilfe des Tenant Managers oder der Tenant Management API konfiguriert. Sie haben folgende Form: • `arn:aws:es:region:account-ID:domain/mydomain/myindex/mytype` • `urn:mysite:es:::mydomain/myindex/mytype` Der Endpunkt muss konfiguriert werden, bevor die Konfigurations-XML übermittelt wird, andernfalls schlägt die Konfiguration mit einem 404-Fehler fehl. Die Urne ist im Zielelement enthalten.	Ja

Anforderungsbeispiele

Dieses Beispiel zeigt die Aktivierung der Suchintegration für einen Bucket. In diesem Beispiel werden die Objektmetadaten für alle Objekte an dasselbe Ziel gesendet.

```
PUT /test1?x-ntap-sg-metadata-notification HTTP/1.1
Date: date
Authorization: authorization string
Host: host

<MetadataNotificationConfiguration>
    <Rule>
        <ID>Rule-1</ID>
        <Status>Enabled</Status>
        <Prefix></Prefix>
        <Destination>
           <Urn>urn:sgws:es:::sgws-notifications/test1/all</Urn>
        </Destination>
    </Rule>
</MetadataNotificationConfiguration>
```

In diesem Beispiel werden Objektmetadaten für Objekte verwendet, die mit dem Präfix `/images` wird an ein Ziel gesendet, während Objektmetadaten für Objekte, die dem Präfix entsprechen `/videos` wird an ein zweites Ziel gesendet.

```
PUT /graphics?x-ntap-sg-metadata-notification HTTP/1.1
Date: date
Authorization: authorization string
Host: host

<MetadataNotificationConfiguration>
    <Rule>
        <ID>Images-rule</ID>
        <Status>Enabled</Status>
        <Prefix>/images</Prefix>
        <Destination>
           <Urn>arn:aws:es:us-east-1:3333333:domain/es-
domain/graphics/imagetype</Urn>
        </Destination>
    </Rule>
    <Rule>
        <ID>Videos-rule</ID>
        <Status>Enabled</Status>
        <Prefix>/videos</Prefix>
        <Destination>
           <Urn>arn:aws:es:us-west-1:22222222:domain/es-
domain/graphics/videotype</Urn>
        </Destination>
    </Rule>
</MetadataNotificationConfiguration>
```

Vom Suchintegrationsdienst generiertes JSON

Wenn Sie den Suchintegrationsdienst für einen Bucket aktivieren, wird jedes Mal, wenn Objektmetadaten oder Tags hinzugefügt, aktualisiert oder gelöscht werden, ein JSON-Dokument generiert und an den Zielendpunkt gesendet.

Dieses Beispiel zeigt ein Beispiel des JSON, das generiert werden könnte, wenn ein Objekt mit dem Schlüssel `SGWS/Tagging.txt` wird in einem Bucket namens erstellt `test` . Der `test` Bucket ist nicht versioniert, also die `versionId` -Tag ist leer.

```
{
  "bucket": "test",
  "key": "SGWS/Tagging.txt",
  "versionId": "",
  "accountId": "86928401983529626822",
  "size": 38,
  "md5": "3d6c7634a85436eee06d43415012855",
  "region":"us-east-1",
  "metadata": {
    "age": "25"
  },
  "tags": {
    "color": "yellow"
  }
}
```

In Metadatenbenachrichtigungen enthaltene Objektmetadaten

In der Tabelle sind alle Felder aufgeführt, die im JSON-Dokument enthalten sind, das an den Zielendpunkt gesendet wird, wenn die Suchintegration aktiviert ist.

Der Dokumentname umfasst den Bucket-Namen, den Objektnamen und die Versions-ID, falls vorhanden.

Typ	Artikelname	Beschreibung
Bucket- und Objektinformationen	Eimer	Name des Buckets
Bucket- und Objektinformationen	Schlüssel	Objektschlüsselname
Bucket- und Objektinformationen	Versions-ID	Objektversion für Objekte in versionierten Buckets
Bucket- und Objektinformationen	Region	Bucket-Region, zum Beispiel `us-east-1`
Systemmetadaten	Größe	Objektgröße (in Bytes), wie sie für einen HTTP-Client sichtbar ist
Systemmetadaten	md5	Objekt-Hash
Benutzermetadaten	Metadaten *`key:value`*	Alle Benutzermetadaten für das Objekt als Schlüssel-Wert-Paare
Schlagwörter	Schlagworte *`key:value`*	Alle für das Objekt definierten Objekt-Tags als Schlüssel-Wert-Paare

Für Tags und Benutzermetadaten übergibt StorageGRID Daten und Zahlen als Zeichenfolgen oder als S3-Ereignisbenachrichtigungen an Elasticsearch. Um Elasticsearch so zu konfigurieren, dass diese Zeichenfolgen als Datumsangaben oder Zahlen interpretiert werden, befolgen Sie die Elasticsearch-Anweisungen zur dynamischen Feldzuordnung und zur Zuordnung von Datumsformaten. Sie müssen die dynamischen Feldzuordnungen im Index aktivieren, bevor Sie den Suchintegrationsdienst konfigurieren. Nachdem ein Dokument indiziert wurde, können Sie die Feldtypen des Dokuments im Index nicht mehr bearbeiten.

Ähnliche Informationen

"Verwenden eines Mandantenkontos"

GET-Speichernutzungsanforderung

Die Anforderung „GET Storage Usage“ gibt Auskunft über die Gesamtmenge des von einem Konto und jedem mit dem Konto verknüpften Bucket verwendeten Speichers.

Die Menge des von einem Konto und seinen Buckets verwendeten Speichers kann durch eine modifizierte ListBuckets-Anfrage mit dem `x-ntap-sg-usage` Abfrageparameter. Die Bucket-Speichernutzung wird getrennt von den vom System verarbeiteten PUT- und DELETE-Anfragen verfolgt. Es kann zu einer gewissen Verzögerung kommen, bevor die Nutzungswerte den erwarteten Werten auf Grundlage der Verarbeitung von Anfragen entsprechen, insbesondere wenn das System stark ausgelastet ist.

Standardmäßig versucht StorageGRID , Nutzungsinformationen mithilfe einer starken globalen Konsistenz abzurufen. Wenn keine starke globale Konsistenz erreicht werden kann, versucht StorageGRID , die Nutzungsinformationen mit einer starken Site-Konsistenz abzurufen.

Sie müssen über die Berechtigung s3:ListAllMyBuckets verfügen oder Root-Kontoinhaber sein, um diesen Vorgang abzuschließen.

Anforderungsbeispiel

```
GET /?x-ntap-sg-usage HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

Antwortbeispiel

Dieses Beispiel zeigt ein Konto mit vier Objekten und 12 Byte Daten in zwei Buckets. Jeder Bucket enthält zwei Objekte und sechs Bytes Daten.

```
HTTP/1.1 200 OK
Date: Sat, 29 Nov 2015 00:49:05 GMT
Connection: KEEP-ALIVE
Server: StorageGRID/10.2.0
x-amz-request-id: 727237123
Content-Length: 427
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<UsageResult xmlns="http://s3.storagegrid.com/doc/2015-02-01">
<CalculationTime>2014-11-19T05:30:11.000000Z</CalculationTime>
<ObjectCount>4</ObjectCount>
<DataBytes>12</DataBytes>
<Buckets>
<Bucket>
<Name>bucket1</Name>
<ObjectCount>2</ObjectCount>
<DataBytes>6</DataBytes>
</Bucket>
<Bucket>
<Name>bucket2</Name>
<ObjectCount>2</ObjectCount>
<DataBytes>6</DataBytes>
</Bucket>
</Buckets>
</UsageResult>
```

Versionierung

Jede gespeicherte Objektversion trägt dazu bei, `ObjectCount` Und `DataBytes` Werte in der Antwort. Löschmarkierungen werden nicht hinzugefügt zum `ObjectCount` gesamt.

Ähnliche Informationen

"Konsistenzwerte"

Veraltete Bucket-Anfragen für Legacy-Compliance

Veraltete Bucket-Anfragen für Legacy-Compliance

Möglicherweise müssen Sie die StorageGRID S3 REST-API verwenden, um Buckets zu verwalten, die mit der alten Compliance-Funktion erstellt wurden.

Compliance-Funktion veraltet

Die StorageGRID Compliance-Funktion, die in früheren StorageGRID Versionen verfügbar war, ist veraltet und wurde durch S3 Object Lock ersetzt.

Wenn Sie zuvor die globale Compliance-Einstellung aktiviert haben, ist die globale S3-Objektsperreinstellung in StorageGRID 11.6 aktiviert. Sie können keine neuen Buckets mehr erstellen, wenn Compliance aktiviert ist. Bei Bedarf können Sie jedoch die StorageGRID S3 REST API verwenden, um vorhandene ältere konforme Buckets zu verwalten.

- "Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren"
- "Objekte mit ILM verwalten"
- "NetApp Knowledge Base: So verwalten Sie ältere konforme Buckets in StorageGRID 11.5"

Veraltete Compliance-Anfragen:

- "Veraltet – PUT-Bucket-Anforderungsänderungen zur Einhaltung der Vorschriften"

  Das XML-Element SGCompliance ist veraltet. Bisher konnten Sie dieses benutzerdefinierte StorageGRID Element in den optionalen XML-Anforderungstext von PUT-Bucket-Anforderungen aufnehmen, um einen konformen Bucket zu erstellen.

- "Veraltet – GET Bucket-Konformität"

  Die GET Bucket-Compliance-Anforderung ist veraltet. Sie können diese Anfrage jedoch weiterhin verwenden, um die aktuell geltenden Compliance-Einstellungen für einen vorhandenen Legacy-Compliant-Bucket zu ermitteln.

- "Veraltet – PUT-Bucket-Konformität"

  Die PUT-Bucket-Compliance-Anforderung ist veraltet. Sie können diese Anfrage jedoch weiterhin verwenden, um die Compliance-Einstellungen für einen vorhandenen Legacy-Compliant-Bucket zu ändern. Sie können beispielsweise einen vorhandenen Bucket auf Legal Hold setzen oder seine Aufbewahrungsdauer verlängern.

Veraltet: CreateBucket-Anforderungsänderungen zur Einhaltung der Vorschriften

Das XML-Element SGCompliance ist veraltet. Bisher konnten Sie dieses benutzerdefinierte StorageGRID Element in den optionalen XML-Anforderungstext von CreateBucket-Anforderungen aufnehmen, um einen konformen Bucket zu erstellen.

> Die StorageGRID Compliance-Funktion, die in früheren StorageGRID Versionen verfügbar war, ist veraltet und wurde durch S3 Object Lock ersetzt. Weitere Einzelheiten finden Sie im Folgenden:
>
> - "Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren"
> - "NetApp Knowledge Base: So verwalten Sie ältere konforme Buckets in StorageGRID 11.5"

Sie können keine neuen Buckets mehr erstellen, wenn Compliance aktiviert ist. Die folgende Fehlermeldung wird zurückgegeben, wenn Sie versuchen, mithilfe der CreateBucket-Anforderungsänderungen für die Konformität einen neuen konformen Bucket zu erstellen:

```
The Compliance feature is deprecated.
Contact your StorageGRID administrator if you need to create new Compliant
buckets.
```

Veraltet: GET Bucket-Compliance-Anforderung

Die GET Bucket-Compliance-Anforderung ist veraltet. Sie können diese Anfrage jedoch weiterhin verwenden, um die aktuell geltenden Compliance-Einstellungen für einen vorhandenen Legacy-Compliant-Bucket zu ermitteln.

> Die StorageGRID Compliance-Funktion, die in früheren StorageGRID Versionen verfügbar war, ist veraltet und wurde durch S3 Object Lock ersetzt. Weitere Einzelheiten finden Sie im Folgenden:
>
> - "Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren"
> - "NetApp Knowledge Base: So verwalten Sie ältere konforme Buckets in StorageGRID 11.5"

Sie müssen über die Berechtigung s3:GetBucketCompliance verfügen oder Root-Kontoinhaber sein, um diesen Vorgang abzuschließen.

Anforderungsbeispiel

Mit dieser Beispielanfrage können Sie die Compliance-Einstellungen für den Bucket mit dem Namen ermitteln. `mybucket` .

```
GET /mybucket/?x-ntap-sg-compliance HTTP/1.1
Date: date
Authorization: authorization string
Host: host
```

Antwortbeispiel

In der Antwort-XML `<SGCompliance>` listet die für den Bucket geltenden Compliance-Einstellungen auf. Diese Beispielantwort zeigt die Compliance-Einstellungen für einen Bucket, in dem jedes Objekt ab dem Zeitpunkt der Aufnahme des Objekts in das Grid ein Jahr lang (525.600 Minuten) aufbewahrt wird. Für diesen Bucket besteht derzeit keine rechtliche Sperre. Jedes Objekt wird nach einem Jahr automatisch gelöscht.

```
HTTP/1.1 200 OK
Date: date
Connection: connection
Server: StorageGRID/11.1.0
x-amz-request-id: request ID
Content-Length: length
Content-Type: application/xml

<SGCompliance>
  <RetentionPeriodMinutes>525600</RetentionPeriodMinutes>
  <LegalHold>false</LegalHold>
  <AutoDelete>true</AutoDelete>
</SGCompliance>
```

Name	Beschreibung
Aufbewahrungsdauer in Minuten	Die Länge der Aufbewahrungsfrist für diesem Bucket hinzugefügte Objekte in Minuten. Die Aufbewahrungsfrist beginnt, wenn das Objekt in das Raster aufgenommen wird.
LegalHold	• True: Dieser Bucket unterliegt derzeit einer rechtlichen Sperre. Objekte in diesem Bucket können erst gelöscht werden, wenn die rechtliche Sperre aufgehoben wird, auch wenn ihre Aufbewahrungsfrist abgelaufen ist. • Falsch: Dieser Bucket unterliegt derzeit keiner rechtlichen Sperre. Objekte in diesem Bucket können gelöscht werden, wenn ihre Aufbewahrungsfrist abgelaufen ist.
AutoDelete	• True: Die Objekte in diesem Bucket werden automatisch gelöscht, wenn ihre Aufbewahrungsfrist abläuft, es sei denn, der Bucket unterliegt einer rechtlichen Sperre. • Falsch: Die Objekte in diesem Bucket werden nicht automatisch gelöscht, wenn die Aufbewahrungsfrist abläuft. Sie müssen diese Objekte manuell löschen, wenn Sie sie löschen müssen.

Fehlerantworten

Wenn der Bucket nicht konform erstellt wurde, lautet der HTTP-Statuscode für die Antwort `404 Not Found`, mit einem S3-Fehlercode von `XNoSuchBucketCompliance`.

Veraltet: PUT Bucket-Compliance-Anforderung

Die PUT-Bucket-Compliance-Anforderung ist veraltet. Sie können diese Anfrage jedoch weiterhin verwenden, um die Compliance-Einstellungen für einen vorhandenen Legacy-Compliant-Bucket zu ändern. Sie können beispielsweise einen vorhandenen Bucket auf Legal Hold setzen oder seine Aufbewahrungsdauer verlängern.

> Die StorageGRID Compliance-Funktion, die in früheren StorageGRID Versionen verfügbar war, ist veraltet und wurde durch S3 Object Lock ersetzt. Weitere Einzelheiten finden Sie im Folgenden:
>
> • "Verwenden Sie die S3 REST API, um S3 Object Lock zu konfigurieren"
> • "NetApp Knowledge Base: So verwalten Sie ältere konforme Buckets in StorageGRID 11.5"

Sie müssen über die Berechtigung s3:PutBucketCompliance verfügen oder Root-Kontobenutzer sein, um diesen Vorgang abzuschließen.

Sie müssen für jedes Feld der Compliance-Einstellungen einen Wert angeben, wenn Sie eine PUT-Bucket-Compliance-Anforderung stellen.

Anforderungsbeispiel

Diese Beispielanforderung ändert die Compliance-Einstellungen für den Bucket mit dem Namen `mybucket`. In diesem Beispiel werden Objekte in `mybucket` werden nun zwei Jahre (1.051.200 Minuten) statt einem Jahr

aufbewahrt, beginnend mit der Aufnahme des Objekts in das Grid. Für diesen Bucket besteht keine rechtliche Sperre. Jedes Objekt wird nach zwei Jahren automatisch gelöscht.

```
PUT /mybucket/?x-ntap-sg-compliance HTTP/1.1
Date: date
Authorization: authorization name
Host: host
Content-Length: 152

<SGCompliance>
  <RetentionPeriodMinutes>1051200</RetentionPeriodMinutes>
  <LegalHold>false</LegalHold>
  <AutoDelete>true</AutoDelete>
</SGCompliance>
```

Name	Beschreibung
Aufbewahrungsdauer in Minuten	Die Länge der Aufbewahrungsfrist für diesem Bucket hinzugefügte Objekte in Minuten. Die Aufbewahrungsfrist beginnt, wenn das Objekt in das Raster aufgenommen wird. **Wichtig** Wenn Sie einen neuen Wert für RetentionPeriodMinutes angeben, müssen Sie einen Wert angeben, der gleich oder größer als die aktuelle Aufbewahrungsdauer des Buckets ist. Nachdem die Aufbewahrungsdauer des Buckets festgelegt wurde, können Sie diesen Wert nicht mehr verringern, sondern nur erhöhen.
LegalHold	• True: Dieser Bucket unterliegt derzeit einer rechtlichen Sperre. Objekte in diesem Bucket können erst gelöscht werden, wenn die rechtliche Sperre aufgehoben wird, auch wenn ihre Aufbewahrungsfrist abgelaufen ist. • Falsch: Dieser Bucket unterliegt derzeit keiner rechtlichen Sperre. Objekte in diesem Bucket können gelöscht werden, wenn ihre Aufbewahrungsfrist abgelaufen ist.
AutoDelete	• True: Die Objekte in diesem Bucket werden automatisch gelöscht, wenn ihre Aufbewahrungsfrist abläuft, es sei denn, der Bucket unterliegt einer rechtlichen Sperre. • Falsch: Die Objekte in diesem Bucket werden nicht automatisch gelöscht, wenn die Aufbewahrungsfrist abläuft. Sie müssen diese Objekte manuell löschen, wenn Sie sie löschen müssen.

Konsistenz für Compliance-Einstellungen

Wenn Sie die Compliance-Einstellungen für einen S3-Bucket mit einer PUT-Bucket-Compliance-Anforderung aktualisieren, versucht StorageGRID , die Metadaten des Buckets im gesamten Grid zu aktualisieren. Standardmäßig verwendet StorageGRID die **starke globale** Konsistenz, um zu gewährleisten, dass alle Rechenzentrumsstandorte und alle Speicherknoten, die Bucket-Metadaten enthalten, für die geänderten Compliance-Einstellungen eine Lese-nach-Schreib-Konsistenz aufweisen.

Wenn StorageGRID die **Starke globale** Konsistenz nicht erreichen kann, weil ein Rechenzentrumsstandort oder mehrere Speicherknoten an einem Standort nicht verfügbar sind, lautet der HTTP-Statuscode für die Antwort `503 Service Unavailable.`

Wenn Sie diese Antwort erhalten, müssen Sie sich an den Grid-Administrator wenden, um sicherzustellen, dass die erforderlichen Speicherdienste so schnell wie möglich bereitgestellt werden. Wenn der Grid-Administrator nicht in der Lage ist, genügend Speicherknoten an jedem Standort verfügbar zu machen, weist Sie der technische Support möglicherweise an, die fehlgeschlagene Anfrage zu wiederholen, indem er die **Strong-Site**-Konsistenz erzwingt.

> Erzwingen Sie niemals die **Strong-Site**-Konsistenz für die PUT-Bucket-Konformität, es sei denn, Sie wurden vom technischen Support dazu aufgefordert und sind sich der möglichen Konsequenzen der Verwendung dieser Ebene bewusst.

Wenn die Konsistenz auf **Strong-Site** reduziert wird, garantiert StorageGRID , dass aktualisierte Compliance-Einstellungen nur für Clientanforderungen innerhalb einer Site eine Read-After-Write-Konsistenz aufweisen. Dies bedeutet, dass das StorageGRID -System vorübergehend mehrere inkonsistente Einstellungen für diesen Bucket haben könnte, bis alle Sites und Storage Nodes verfügbar sind. Die inkonsistenten Einstellungen können zu unerwartetem und unerwünschtem Verhalten führen. Wenn Sie beispielsweise einen Bucket einer rechtlichen Sperre unterziehen und eine geringere Konsistenz erzwingen, bleiben die vorherigen Compliance-Einstellungen des Buckets (d. h. die rechtliche Sperre) an einigen Rechenzentrumsstandorten möglicherweise weiterhin wirksam. Dies hat zur Folge, dass Objekte, die Ihrer Meinung nach rechtlich gesperrt sind, nach Ablauf ihrer Aufbewahrungsfrist möglicherweise gelöscht werden, entweder durch den Benutzer oder durch AutoDelete (sofern aktiviert).

Um die Verwendung der **Strong-site**-Konsistenz zu erzwingen, stellen Sie die PUT Bucket-Compliance-Anforderung erneut aus und schließen Sie die `Consistency-Control` HTTP-Anforderungsheader, wie folgt:

```
PUT /mybucket/?x-ntap-sg-compliance HTTP/1.1
Consistency-Control: strong-site
```

Fehlerantworten

- Wenn der Bucket nicht konform erstellt wurde, lautet der HTTP-Statuscode für die Antwort `404 Not Found` .
- Wenn `RetentionPeriodMinutes` in der Anfrage kleiner ist als die aktuelle Aufbewahrungsdauer des Buckets, lautet der HTTP-Statuscode `400 Bad Request` .

Ähnliche Informationen

"Veraltet: PUT-Bucket-Anforderungsänderungen zur Einhaltung der Vorschriften"

Bucket- und Gruppenzugriffsrichtlinien

Verwenden Sie Bucket- und Gruppenzugriffsrichtlinien

StorageGRID verwendet die Richtliniensprache von Amazon Web Services (AWS), um S3-Mietern die Kontrolle über den Zugriff auf Buckets und Objekte in diesen Buckets zu ermöglichen. Das StorageGRID -System implementiert eine Teilmenge der S3 REST API-Richtliniensprache. Zugriffsrichtlinien für die S3-API sind in JSON geschrieben.

Übersicht über die Zugriffsrichtlinie

StorageGRID unterstützt zwei Arten von Zugriffsrichtlinien.

- **Bucket-Richtlinien**, die mithilfe der S3-API-Operationen GetBucketPolicy, PutBucketPolicy und DeleteBucketPolicy oder der Tenant Manager- oder Tenant Management-API verwaltet werden. Bucket-Richtlinien sind an Buckets angehängt und daher so konfiguriert, dass sie den Zugriff von Benutzern im Bucket-Eigentümerkonto oder anderen Konten auf den Bucket und die darin enthaltenen Objekte steuern. Eine Bucket-Richtlinie gilt nur für einen Bucket und möglicherweise für mehrere Gruppen.
- **Gruppenrichtlinien**, die mithilfe des Tenant Managers oder der Tenant Management API konfiguriert werden. Gruppenrichtlinien sind einer Gruppe im Konto zugeordnet und daher so konfiguriert, dass diese Gruppe auf bestimmte Ressourcen zugreifen kann, die diesem Konto gehören. Eine Gruppenrichtlinie gilt nur für eine Gruppe und möglicherweise mehrere Buckets.

Es gibt keinen Unterschied in der Priorität zwischen Gruppen- und Bucket-Richtlinien.

StorageGRID Bucket- und Gruppenrichtlinien folgen einer bestimmten, von Amazon definierten Grammatik. Innerhalb jeder Richtlinie befindet sich ein Array von Richtlinienanweisungen und jede Anweisung enthält die folgenden Elemente:

- Anweisungs-ID (Sid) (optional)
- Wirkung
- Auftraggeber/NichtAuftraggeber
- Ressource/NichtRessource
- Aktion/NichtAktion
- Bedingung (optional)

Richtlinienanweisungen werden mithilfe dieser Struktur erstellt, um Berechtigungen anzugeben: Gewähren Sie <Effekt>, um <Principal> die Ausführung von <Aktion> auf <Ressource> zu erlauben/verweigern, wenn <Bedingung> zutrifft.

Jedes Richtlinienelement wird für eine bestimmte Funktion verwendet:

Element	Beschreibung
Sid	Das Sid-Element ist optional. Die Sid dient lediglich als Beschreibung für den Benutzer. Es wird gespeichert, aber nicht vom StorageGRID -System interpretiert.
Wirkung	Verwenden Sie das Effect-Element, um festzulegen, ob die angegebenen Vorgänge zulässig oder verweigert werden. Sie müssen Vorgänge, die Sie für Buckets oder Objekte zulassen (oder verweigern), mithilfe der unterstützten Schlüsselwörter des Aktionselements identifizieren.

Element	Beschreibung
Auftraggeber/NichtAuftraggeber	Sie können Benutzern, Gruppen und Konten den Zugriff auf bestimmte Ressourcen und die Ausführung bestimmter Aktionen gestatten. Wenn in der Anfrage keine S3-Signatur enthalten ist, wird der anonyme Zugriff durch Angabe des Platzhalterzeichens (*) als Prinzipal zugelassen. Standardmäßig hat nur der Konto-Root Zugriff auf die Ressourcen, die dem Konto gehören. Sie müssen nur das Principal-Element in einer Bucket-Richtlinie angeben. Bei Gruppenrichtlinien ist die Gruppe, an die die Richtlinie angehängt ist, das implizite Principal-Element.
Ressource/NichtRessource	Das Ressourcenelement identifiziert Buckets und Objekte. Sie können Berechtigungen für Buckets und Objekte erteilen oder verweigern, indem Sie den Amazon Resource Name (ARN) zur Identifizierung der Ressource verwenden.
Aktion/NichtAktion	Die Elemente „Aktion“ und „Effekt“ sind die beiden Komponenten von Berechtigungen. Wenn eine Gruppe eine Ressource anfordert, wird ihr der Zugriff auf die Ressource entweder gewährt oder verweigert. Der Zugriff wird verweigert, sofern Sie keine ausdrücklichen Berechtigungen erteilen. Sie können jedoch eine durch eine andere Richtlinie erteilte Berechtigung durch eine explizite Verweigerung außer Kraft setzen.
Zustand	Das Bedingungselement ist optional. Mithilfe von Bedingungen können Sie Ausdrücke erstellen, um zu bestimmen, wann eine Richtlinie angewendet werden soll.

Im Aktionselement können Sie das Platzhalterzeichen (*) verwenden, um alle Vorgänge oder eine Teilmenge von Vorgängen anzugeben. Diese Aktion entspricht beispielsweise Berechtigungen wie s3:GetObject, s3:PutObject und s3:DeleteObject.

```
s3:*Object
```

Im Ressourcenelement können Sie die Platzhalterzeichen (*) und (?) verwenden. Während das Sternchen (*) 0 oder mehr Zeichen entspricht, entspricht das Fragezeichen (?) einem beliebigen einzelnen Zeichen.

Im Principal-Element werden Platzhalterzeichen nur zum Festlegen des anonymen Zugriffs unterstützt, der jedem die Berechtigung erteilt. Beispielsweise legen Sie das Platzhalterzeichen (*) als Hauptwert fest.

```
"Principal":"*"
```

```
"Principal":{"AWS":"*"}
```

Im folgenden Beispiel verwendet die Anweisung die Elemente „Effect“, „Principal“, „Action“ und „Resource“. Dieses Beispiel zeigt eine vollständige Bucket-Richtlinienanweisung, die den Effekt "Zulassen" verwendet, um

den Principals, der Admin-Gruppe `federated-group/admin` und die Finanzgruppe `federated-group/finance` , Berechtigungen zum Ausführen der Aktion `s3:ListBucket` auf dem Eimer namens `mybucket` und die Aktion `s3:GetObject` auf allen Objekten in diesem Bucket.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::27233906934684427525:federated-group/admin",
          "arn:aws:iam::27233906934684427525:federated-group/finance"
        ]
      },
      "Action": [
        "s3:ListBucket",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::mybucket",
        "arn:aws:s3:::mybucket/*"
      ]
    }
  ]
}
```

Die Bucket-Richtlinie hat eine Größenbeschränkung von 20.480 Bytes und die Gruppenrichtlinie eine Größenbeschränkung von 5.120 Bytes.

Konsistenz für Richtlinien

Standardmäßig sind alle Aktualisierungen, die Sie an Gruppenrichtlinien vornehmen, letztendlich konsistent. Wenn eine Gruppenrichtlinie konsistent wird, kann es aufgrund der Richtlinienzwischenspeicherung weitere 15 Minuten dauern, bis die Änderungen wirksam werden. Standardmäßig sind alle Aktualisierungen, die Sie an Bucket-Richtlinien vornehmen, streng konsistent.

Bei Bedarf können Sie die Konsistenzgarantien für Bucket-Richtlinienaktualisierungen ändern. Beispielsweise möchten Sie möglicherweise, dass eine Änderung an einer Bucket-Richtlinie während eines Site-Ausfalls verfügbar ist.

In diesem Fall können Sie entweder die `Consistency-Control` Header in der PutBucketPolicy-Anforderung, oder Sie können die PUT Bucket-Konsistenzanforderung verwenden. Wenn eine Bucket-Richtlinie konsistent wird, kann es aufgrund der Richtlinienzwischenspeicherung weitere 8 Sekunden dauern, bis die Änderungen wirksam werden.

Wenn Sie die Konsistenz auf einen anderen Wert einstellen, um eine vorübergehende Situation zu umgehen, denken Sie daran, die Einstellung auf Bucket-Ebene wieder auf den ursprünglichen Wert zurückzusetzen, wenn Sie fertig sind. Andernfalls verwenden alle zukünftigen Bucket-Anfragen die geänderte Einstellung.

Verwenden Sie ARN in Richtlinienanweisungen

In Richtlinienanweisungen wird die ARN in den Elementen „Principal“ und „Resource“ verwendet.

- Verwenden Sie diese Syntax, um die S3-Ressourcen-ARN anzugeben:

```
arn:aws:s3:::bucket-name
arn:aws:s3:::bucket-name/object_key
```

- Verwenden Sie diese Syntax, um die ARN der Identitätsressource (Benutzer und Gruppen) anzugeben:

```
arn:aws:iam::account_id:root
arn:aws:iam::account_id:user/user_name
arn:aws:iam::account_id:group/group_name
arn:aws:iam::account_id:federated-user/user_name
arn:aws:iam::account_id:federated-group/group_name
```

Weitere Überlegungen:

- Sie können das Sternchen (*) als Platzhalter verwenden, um null oder mehr Zeichen im Objektschlüssel abzugleichen.
- Internationale Zeichen, die im Objektschlüssel angegeben werden können, sollten mit JSON UTF-8 oder mit JSON \u-Escapesequenzen codiert werden. Prozentkodierung wird nicht unterstützt.

  "RFC 2141 URN-Syntax"

  Der HTTP-Anforderungstext für den PutBucketPolicy-Vorgang muss mit charset=UTF-8 codiert sein.

Angeben von Ressourcen in einer Richtlinie

In Richtlinienanweisungen können Sie das Ressourcenelement verwenden, um den Bucket oder das Objekt anzugeben, für das Berechtigungen erteilt oder verweigert werden.

- Jede Richtlinienanweisung erfordert ein Ressourcenelement. In einer Richtlinie werden Ressourcen durch das Element gekennzeichnet `Resource` oder alternativ `NotResource` zum Ausschluss.
- Sie geben Ressourcen mit einer S3-Ressourcen-ARN an. Beispiel:

```
"Resource": "arn:aws:s3:::mybucket/*"
```

- Sie können auch Richtlinienvariablen innerhalb des Objektschlüssels verwenden. Beispiel:

```
"Resource": "arn:aws:s3:::mybucket/home/${aws:username}/*"
```

- Der Ressourcenwert kann einen Bucket angeben, der beim Erstellen einer Gruppenrichtlinie noch nicht vorhanden ist.

Angeben von Prinzipalen in einer Richtlinie

Verwenden Sie das Principal-Element, um den Benutzer, die Gruppe oder das Mandantenkonto zu identifizieren, dem durch die Richtlinienanweisung der Zugriff auf die Ressource gestattet bzw. verweigert wird.

- Jede Richtlinienanweisung in einer Bucket-Richtlinie muss ein Principal-Element enthalten. Richtlinienanweisungen in einer Gruppenrichtlinie benötigen das Principal-Element nicht, da die Gruppe als Auftraggeber verstanden wird.
- In einer Richtlinie werden Auftraggeber durch das Element „Principal“ oder alternativ „NotPrincipal“ zum Ausschluss gekennzeichnet.
- Kontobasierte Identitäten müssen mithilfe einer ID oder einer ARN angegeben werden:

```
"Principal": { "AWS": "account_id"}
"Principal": { "AWS": "identity_arn" }
```

- In diesem Beispiel wird die Mandantenkonto-ID 27233906934684427525 verwendet, die das Stammkonto und alle Benutzer im Konto umfasst:

```
 "Principal": { "AWS": "27233906934684427525" }
```

- Sie können nur das Stammkonto angeben:

```
"Principal": { "AWS": "arn:aws:iam::27233906934684427525:root" }
```

- Sie können einen bestimmten Verbundbenutzer („Alex“) angeben:

```
"Principal": { "AWS": "arn:aws:iam::27233906934684427525:federated-
user/Alex" }
```

- Sie können eine bestimmte föderierte Gruppe („Manager“) angeben:

```
"Principal": { "AWS": "arn:aws:iam::27233906934684427525:federated-
group/Managers"  }
```

- Sie können einen anonymen Auftraggeber angeben:

```
"Principal": "*"
```

- Um Mehrdeutigkeiten zu vermeiden, können Sie anstelle des Benutzernamens die Benutzer-UUID verwenden:

```
arn:aws:iam::27233906934684427525:user-uuid/de305d54-75b4-431b-adb2-eb6b9e546013
```

  Nehmen wir beispielsweise an, Alex verlässt die Organisation und der Benutzername `Alex` wird gelöscht. Wenn ein neuer Alex in die Organisation eintritt und ihm die gleiche `Alex` Benutzernamen, könnte der neue Benutzer unbeabsichtigt die dem ursprünglichen Benutzer erteilten Berechtigungen erben.

- Der Prinzipalwert kann einen Gruppen-/Benutzernamen angeben, der beim Erstellen einer Bucket-Richtlinie noch nicht vorhanden ist.

Festlegen von Berechtigungen in einer Richtlinie

In einer Richtlinie wird das Aktionselement verwendet, um Berechtigungen für eine Ressource zuzulassen/zu verweigern. Es gibt eine Reihe von Berechtigungen, die Sie in einer Richtlinie angeben können. Diese werden durch das Element „Action“ oder alternativ „NotAction“ zum Ausschluss gekennzeichnet. Jedes dieser Elemente ist bestimmten S3 REST-API-Operationen zugeordnet.

In den Tabellen sind die Berechtigungen aufgeführt, die für Buckets gelten, und die Berechtigungen, die für Objekte gelten.

> Amazon S3 verwendet jetzt die Berechtigung s3:PutReplicationConfiguration sowohl für die Aktionen PutBucketReplication als auch DeleteBucketReplication. StorageGRID verwendet für jede Aktion separate Berechtigungen, was der ursprünglichen Amazon S3-Spezifikation entspricht.

> Ein Löschen wird ausgeführt, wenn ein Put zum Überschreiben eines vorhandenen Werts verwendet wird.

Berechtigungen, die für Buckets gelten

Berechtigungen	S3 REST API-Operationen	Benutzerdefiniert für StorageGRID
s3:Bucket erstellen	Bucket erstellen	Ja. **Hinweis**: Nur in Gruppenrichtlinien verwenden.
s3:Bucket löschen	Bucket löschen	
s3:DeleteBucketMetadataNotification	Konfiguration der Benachrichtigung über DELETE-Bucket-Metadaten	Ja

Berechtigungen	S3 REST API-Operationen	Benutzerdefiniert für StorageGRID
s3:DeleteBucketPolicy	DeleteBucketPolicy	
s3:Replikationskonfiguration löschen	DeleteBucketReplication	Ja, separate Berechtigungen für PUT und DELETE
s3:GetBucketAcl	GetBucketAcl	
s3:GetBucketCompliance	GET Bucket-Konformität (veraltet)	Ja
s3:GetBucketConsistency	GET Bucket-Konsistenz	Ja
s3:GetBucketCORS	GetBucketCors	
s3:GetEncryptionConfiguration	GetBucketEncryption	
s3:GetBucketLastAccessTime	GET Bucket – Letzte Zugriffszeit	Ja
s3:GetBucketLocation	BucketLocation abrufen	
s3:GetBucketMetadataNotification	GET Bucket-Metadaten-Benachrichtigungskonfiguration	Ja
s3:GetBucketNotification	GetBucketNotificationConfiguration	
s3:GetBucketObjectLockConfiguration	GetObjectLockConfiguration	
s3:GetBucketPolicy	GetBucketPolicy	
s3:GetBucketTagging	GetBucketTagging	
s3:GetBucketVersioning	GetBucketVersioning	
s3:GetLifecycleConfiguration	GetBucketLifecycleConfiguration	
s3:GetReplicationConfiguration	GetBucketReplication	
s3:ListeAlleMeineBuckets	• Buckets auflisten • GET-Speichernutzung	Ja, für die GET-Speichernutzung. **Hinweis**: Nur in Gruppenrichtlinien verwenden.

Berechtigungen	S3 REST API-Operationen	Benutzerdefiniert für StorageGRID
s3:ListBucket	• ListObjects • Kopfeimer • RestoreObject	
s3:ListBucketMultipartUploads	• ListMultipartUploads • RestoreObject	
s3:ListBucketVersions	GET Bucket-Versionen	
s3:PutBucketCompliance	PUT-Bucket-Konformität (veraltet)	Ja
s3:PutBucketConsistency	PUT Bucket-Konsistenz	Ja
s3:PutBucketCORS	• DeleteBucketCors† • PutBucketCors	
s3:PutEncryptionConfiguration	• DeleteBucketEncryption • PutBucketEncryption	
s3:PutBucketLastAccessTime	PUT Bucket: Letzte Zugriffszeit	Ja
s3:PutBucketMetadataNotification	Konfiguration der Benachrichtigung über PUT-Bucket-Metadaten	Ja
s3:PutBucketNotification	PutBucketNotificationConfiguration	
s3:PutBucketObjectLockConfiguration	• CreateBucket mit dem `x-amz-bucket-object-lock-enabled: true` Anforderungsheader (erfordert auch die Berechtigung s3:CreateBucket) • PutObjectLockConfiguration	
s3:PutBucketPolicy	PutBucketPolicy	
s3:PutBucketTagging	• BucketTagging löschen† • PutBucketTagging	
s3:PutBucketVersioning	PutBucketVersioning	

Berechtigungen	S3 REST API-Operationen	Benutzerdefiniert für StorageGRID
s3:PutLifecycleConfiguration	• DeleteBucketLifecycle† • PutBucketLifecycleConfiguration	
s3:PutReplicationConfiguration	PutBucketReplication	Ja, separate Berechtigungen für PUT und DELETE

Berechtigungen, die für Objekte gelten

Berechtigungen	S3 REST API-Operationen	Benutzerdefiniert für StorageGRID
s3:AbortMultipartUpload	• AbortMultipartUpload • RestoreObject	
s3:BypassGovernanceRetention	• Objekt löschen • Objekte löschen • PutObjectRetention	
s3:Objekt löschen	• Objekt löschen • Objekte löschen • RestoreObject	
s3:DeleteObjectTagging	DeleteObjectTagging	
s3:DeleteObjectVersionTagging	DeleteObjectTagging (eine bestimmte Version des Objekts)	
s3:DeleteObjectVersion	DeleteObject (eine bestimmte Version des Objekts)	
s3:GetObject	• GetObject • HeadObject • RestoreObject • SelectObjectContent	
s3:GetObjectAcl	GetObjectAcl	
s3:GetObjectLegalHold	GetObjectLegalHold	
s3:GetObjectRetention	GetObjectRetention	

Berechtigungen	S3 REST API-Operationen	Benutzerdefiniert für StorageGRID
s3:GetObjectTagging	GetObjectTagging	
s3:GetObjectVersionTagging	GetObjectTagging (eine bestimmte Version des Objekts)	
s3:GetObjectVersion	GetObject (eine bestimmte Version des Objekts)	
s3:ListMultipartUploadParts	ListParts, RestoreObject	
s3:PutObject	• PutObject • Objekt kopieren • RestoreObject • CreateMultipartUpload • CompleteMultipartUpload • UploadPart • UploadPartCopy	
s3:PutObjectLegalHold	PutObjectLegalHold	
s3:PutObjectRetention	PutObjectRetention	
s3:PutObjectTagging	PutObjectTagging	
s3:PutObjectVersionTagging	PutObjectTagging (eine bestimmte Version des Objekts)	
s3:PutOverwriteObject	• PutObject • Objekt kopieren • PutObjectTagging • DeleteObjectTagging • CompleteMultipartUpload	Ja
s3:RestoreObject	RestoreObject	

PutOverwriteObject-Berechtigung verwenden

Die Berechtigung s3:PutOverwriteObject ist eine benutzerdefinierte StorageGRID Berechtigung, die für Vorgänge gilt, die Objekte erstellen oder aktualisieren. Die Einstellung dieser Berechtigung bestimmt, ob der Client die Daten, benutzerdefinierten Metadaten oder S3-Objektmarkierungen eines Objekts überschreiben kann.

Mögliche Einstellungen für diese Berechtigung sind:

- **Zulassen**: Der Client kann ein Objekt überschreiben. Dies ist die Standardeinstellung.
- **Ablehnen**: Der Client kann ein Objekt nicht überschreiben. Wenn die Berechtigung „PutOverwriteObject“ auf „Verweigern“ gesetzt ist, funktioniert sie wie folgt:
  - Wenn ein vorhandenes Objekt am gleichen Pfad gefunden wird:
    - Die Daten, benutzerdefinierten Metadaten oder S3-Objektmarkierungen des Objekts können nicht überschrieben werden.
    - Alle laufenden Aufnahmevorgänge werden abgebrochen und ein Fehler zurückgegeben.
    - Wenn die S3-Versionierung aktiviert ist, verhindert die Einstellung „Verweigern“, dass PutObjectTagging- oder DeleteObjectTagging-Vorgänge das TagSet für ein Objekt und seine nicht aktuellen Versionen ändern.
  - Wenn ein vorhandenes Objekt nicht gefunden wird, hat diese Berechtigung keine Wirkung.
- Wenn diese Berechtigung nicht vorhanden ist, ist die Wirkung dieselbe, als ob „Zulassen“ gesetzt wäre.

> Wenn die aktuelle S3-Richtlinie das Überschreiben zulässt und die Berechtigung „PutOverwriteObject“ auf „Verweigern“ gesetzt ist, kann der Client die Daten, benutzerdefinierten Metadaten oder Objektmarkierungen eines Objekts nicht überschreiben. Wenn außerdem das Kontrollkästchen **Client-Änderung verhindern** aktiviert ist (**KONFIGURATION** > **Sicherheitseinstellungen** > **Netzwerk und Objekte**), überschreibt diese Einstellung die Einstellung der Berechtigung „PutOverwriteObject“.

Bedingungen in einer Richtlinie angeben

Bedingungen definieren, wann eine Richtlinie in Kraft tritt. Bedingungen bestehen aus Operatoren und Schlüssel-Wert-Paaren.

Bedingungen verwenden Schlüssel-Wert-Paare zur Auswertung. Ein Bedingungselement kann mehrere Bedingungen enthalten und jede Bedingung kann mehrere Schlüssel-Wert-Paare enthalten. Der Bedingungsblock verwendet das folgende Format:

```
Condition: {
     condition_type: {
          condition_key: condition_values
```

Im folgenden Beispiel verwendet die Bedingung „IpAddress“ den Bedingungsschlüssel „SourceIp“.

```
"Condition": {
    "IpAddress": {
      "aws:SourceIp": "54.240.143.0/24"
        ...
},
        ...
```

Unterstützte Bedingungsoperatoren

Bedingungsoperatoren werden wie folgt kategorisiert:

- Zeichenfolge
- Numerisch
- Boolescher Wert
- IP-Adresse
- Nullprüfung

Bedingungsoperatoren	Beschreibung
StringEquals	Vergleicht einen Schlüssel mit einem Zeichenfolgenwert auf Basis einer genauen Übereinstimmung (Groß-/Kleinschreibung beachten).
StringNotEquals	Vergleicht einen Schlüssel mit einem Zeichenfolgenwert basierend auf negierter Übereinstimmung (Groß-/Kleinschreibung beachten).
StringEqualsIgnoreCase	Vergleicht einen Schlüssel mit einem Zeichenfolgenwert basierend auf einer genauen Übereinstimmung (Groß-/Kleinschreibung wird ignoriert).
StringNotEqualsIgnoreCase	Vergleicht einen Schlüssel mit einem Zeichenfolgenwert basierend auf negierter Übereinstimmung (Groß-/Kleinschreibung wird ignoriert).
StringLike	Vergleicht einen Schlüssel mit einem Zeichenfolgenwert auf Basis einer genauen Übereinstimmung (Groß-/Kleinschreibung beachten). Kann die Platzhalterzeichen * und ? enthalten.
StringNotLike	Vergleicht einen Schlüssel mit einem Zeichenfolgenwert basierend auf negierter Übereinstimmung (Groß-/Kleinschreibung beachten). Kann die Platzhalterzeichen * und ? enthalten.
NumericEquals	Vergleicht einen Schlüssel mit einem numerischen Wert auf Basis einer exakten Übereinstimmung.
NumericNotEquals	Vergleicht einen Schlüssel mit einem numerischen Wert basierend auf negierter Übereinstimmung.
NumerischGrößerAls	Vergleicht einen Schlüssel mit einem numerischen Wert basierend auf einer „Größer-als“-Übereinstimmung.
NumerischGrößerAlsGleich	Vergleicht einen Schlüssel mit einem numerischen Wert basierend auf der Übereinstimmung „größer als oder gleich“.
NumericLessThan	Vergleicht einen Schlüssel mit einem numerischen Wert basierend auf einer „kleiner als“-Übereinstimmung.

Bedingungsoperatoren	Beschreibung
NumerischKleinerAlsGleich	Vergleicht einen Schlüssel mit einem numerischen Wert basierend auf der Übereinstimmung „kleiner als oder gleich“.
Bool	Vergleicht einen Schlüssel mit einem Booleschen Wert basierend auf der Übereinstimmung „wahr oder falsch“.
IP-Adresse	Vergleicht einen Schlüssel mit einer IP-Adresse oder einem IP-Adressbereich.
NotIpAddress	Vergleicht einen Schlüssel mit einer IP-Adresse oder einem IP-Adressbereich basierend auf negierter Übereinstimmung.
Null	Überprüft, ob im aktuellen Anforderungskontext ein Bedingungsschlüssel vorhanden ist.

Unterstützte Bedingungsschlüssel

Bedingungsschlüssel	Aktionen	Beschreibung
aws:SourceIp	IP-Betreiber	Wird mit der IP-Adresse verglichen, von der die Anfrage gesendet wurde. Kann für Bucket- oder Objektoperationen verwendet werden. **Hinweis:** Wenn die S3-Anforderung über den Load Balancer-Dienst auf Admin-Knoten und Gateway-Knoten gesendet wurde, wird dies mit der IP-Adresse vor dem Load Balancer-Dienst verglichen. **Hinweis**: Wenn ein nicht transparenter Load Balancer eines Drittanbieters verwendet wird, wird dies mit der IP-Adresse dieses Load Balancers verglichen. Beliebig `X-Forwarded-For` Header wird ignoriert, da seine Gültigkeit nicht festgestellt werden kann.
aws:Benutzername	Ressource/Identität	Wird mit dem Benutzernamen des Absenders verglichen, von dem die Anfrage gesendet wurde. Kann für Bucket- oder Objektoperationen verwendet werden.
s3:Trennzeichen	s3:ListBucket und s3:ListBucketVersions-Berechtigungen	Wird mit dem in einer ListObjects- oder ListObjectVersions-Anforderung angegebenen Trennzeichenparameter verglichen.

Bedingungsschlüssel	Aktionen	Beschreibung
s3:ExistingObjectTag/<Tag-Schlüssel>	s3:DeleteObjectTagging s3:DeleteObjectVersionTagging s3:GetObject s3:GetObjectAcl 3:GetObjectTagging s3:GetObjectVersion s3:GetObjectVersionAcl s3:GetObjectVersionTagging s3:PutObjectAcl s3:PutObjectTagging s3:PutObjectVersionAcl s3:PutObjectVersionTagging	Erfordert, dass das vorhandene Objekt über den spezifischen Tag-Schlüssel und -Wert verfügt.
s3:max-Schlüssel	s3:ListBucket und s3:ListBucketVersions-Berechtigungen	Wird mit dem in einer ListObjects- oder ListObjectVersions-Anforderung angegebenen Max-Keys-Parameter verglichen.
s3:Objektsperre-verbleibende-Aufbewahrungstage	s3:PutObject	Vergleicht mit dem Aufbewahrungsdatum, das in der `x-amz-object-lock-retain-until-date` Anforderungsheader oder berechnet aus der Standardaufbewahrungsdauer des Buckets, um sicherzustellen, dass diese Werte innerhalb des zulässigen Bereichs für die folgenden Anforderungen liegen: • PutObject • Objekt kopieren • CreateMultipartUpload
s3:Objektsperre-verbleibende-Aufbewahrungstage	s3:PutObjectRetention	Vergleicht mit dem in der PutObjectRetention-Anforderung angegebenen Aufbewahrungsdatum, um sicherzustellen, dass es innerhalb des zulässigen Bereichs liegt.

Bedingungsschlüssel	Aktionen	Beschreibung
s3:Präfix	s3:ListBucket und s3:ListBucketVersions-Berechtigungen	Wird mit dem in einer ListObjects- oder ListObjectVersions-Anforderung angegebenen Präfixparameter verglichen.
s3:RequestObjectTag/<Tag-Schlüssel>	s3:PutObject s3:PutObjectTagging s3:PutObjectVersionTagging	Erfordert einen bestimmten Tag-Schlüssel und -Wert, wenn die Objektanforderung Tagging enthält.

Angeben von Variablen in einer Richtlinie

Sie können Variablen in Richtlinien verwenden, um Richtlinieninformationen einzufügen, wenn diese verfügbar sind. Sie können Richtlinienvariablen in der `Resource` Element und in Stringvergleichen im `Condition` Element.

In diesem Beispiel ist die Variable `${aws:username}` ist Teil des Ressourcenelements:

```
"Resource": "arn:aws:s3:::bucket-name/home/${aws:username}/*"
```

In diesem Beispiel ist die Variable `${aws:username}` ist Teil des Bedingungswerts im Bedingungsblock:

```
"Condition": {
    "StringLike": {
      "s3:prefix": "${aws:username}/*"
        ...
},
        ...
```

Variable	Beschreibung
`${aws:SourceIp}`	Verwendet den SourceIp-Schlüssel als bereitgestellte Variable.
`${aws:username}`	Verwendet den Benutzernamenschlüssel als bereitgestellte Variable.
`${s3:prefix}`	Verwendet den dienstspezifischen Präfixschlüssel als bereitgestellte Variable.
`${s3:max-keys}`	Verwendet den dienstspezifischen Max-Keys-Schlüssel als bereitgestellte Variable.

Variable	Beschreibung
`${*}`	Sonderzeichen. Verwendet das Zeichen als wörtliches *-Zeichen.
`${?}`	Sonderzeichen. Verwendet das Zeichen als wörtliches ?-Zeichen.
`${$}`	Sonderzeichen. Verwendet das Zeichen als wörtliches $-Zeichen.

Erstellen Sie Richtlinien, die eine besondere Behandlung erfordern

Manchmal kann eine Richtlinie Berechtigungen erteilen, die eine Gefahr für die Sicherheit oder den laufenden Betrieb darstellen, wie etwa das Sperren des Root-Benutzers des Kontos. Die StorageGRID S3 REST-API-Implementierung ist bei der Richtlinienvalidierung weniger restriktiv als Amazon, bei der Richtlinienauswertung jedoch ebenso streng.

Richtlinienbeschreibung	Richtlinientyp	Amazon-Verhalten	StorageGRID -Verhalten
Verweigern Sie sich selbst alle Berechtigungen für das Root-Konto	Eimer	Gültig und erzwungen, aber das Root-Benutzerkonto behält die Berechtigung für alle S3-Bucket-Richtlinienvorgänge	Dasselbe
Sich selbst alle Berechtigungen für Benutzer/Gruppe verweigern	Gruppe	Gültig und durchgesetzt	Dasselbe
Erteilen Sie einer fremden Kontogruppe alle Berechtigungen	Eimer	Ungültiger Auftraggeber	Gültig, aber Berechtigungen für alle S3-Bucket-Richtlinienvorgänge geben einen 405-Methodenfehler zurück, wenn sie durch eine Richtlinie erlaubt sind
Erteilen Sie einem fremden Root- oder Benutzerkonto alle Berechtigungen	Eimer	Gültig, aber Berechtigungen für alle S3-Bucket-Richtlinienvorgänge geben einen 405-Methodenfehler zurück, wenn sie durch eine Richtlinie erlaubt sind	Dasselbe
Jedem die Berechtigung für alle Aktionen erteilen	Eimer	Gültig, aber Berechtigungen für alle S3-Bucket-Richtlinienvorgänge geben einen 405-Methode nicht zulässig-Fehler für das Stammkonto und die Benutzer des Fremdkontos zurück	Dasselbe

Richtlinienbeschreibung	Richtlinientyp	Amazon-Verhalten	StorageGRID -Verhalten
Allen die Berechtigung für alle Aktionen verweigern	Eimer	Gültig und erzwungen, aber das Root-Benutzerkonto behält die Berechtigung für alle S3-Bucket-Richtlinienvorgänge	Dasselbe
Der Auftraggeber ist ein nicht vorhandener Benutzer oder eine nicht vorhandene Gruppe.	Eimer	Ungültiger Auftraggeber	Gültig
Ressource ist ein nicht vorhandener S3-Bucket	Gruppe	Gültig	Dasselbe
Principal ist eine lokale Gruppe	Eimer	Ungültiger Auftraggeber	Gültig
Die Richtlinie erteilt Nichtbesitzerkonten (einschließlich anonymer Konten) die Berechtigung, Objekte abzulegen.	Eimer	Gültig. Objekte sind Eigentum des Erstellerkontos und die Bucket-Richtlinie gilt nicht. Das Erstellerkonto muss mithilfe von Objekt-ACLs Zugriffsberechtigungen für das Objekt erteilen.	Gültig. Objekte sind Eigentum des Bucket-Eigentümerkontos. Es gilt die Bucket-Richtlinie.

WORM-Schutz (Write-Once-Read-Many)

Sie können WORM-Buckets (Write-Once-Read-Many) erstellen, um Daten, benutzerdefinierte Objektmetadaten und S3-Objekt-Tagging zu schützen. Sie konfigurieren die WORM-Buckets, um die Erstellung neuer Objekte zu ermöglichen und das Überschreiben oder Löschen vorhandener Inhalte zu verhindern. Verwenden Sie einen der hier beschriebenen Ansätze.

Um sicherzustellen, dass Überschreibungen immer verweigert werden, können Sie:

- Gehen Sie im Grid Manager zu **KONFIGURATION** > **Sicherheit** > **Sicherheitseinstellungen** > **Netzwerk und Objekte** und aktivieren Sie das Kontrollkästchen **Client-Änderung verhindern**.
- Wenden Sie die folgenden Regeln und S3-Richtlinien an:
  - Fügen Sie der S3-Richtlinie eine PutOverwriteObject DENY-Operation hinzu.
  - Fügen Sie der S3-Richtlinie eine DeleteObject DENY-Operation hinzu.
  - Fügen Sie der S3-Richtlinie eine PutObject ALLOW-Operation hinzu.

> Das Festlegen von „DeleteObject“ auf „DENY“ in einer S3-Richtlinie verhindert nicht, dass ILM Objekte löscht, wenn eine Regel wie „Null Kopien nach 30 Tagen“ vorhanden ist.

Selbst wenn alle diese Regeln und Richtlinien angewendet werden, schützen sie nicht vor gleichzeitigen Schreibvorgängen (siehe Situation A). Sie schützen vor sequenziellen Überschreibungen (siehe Situation B).

Situation A: Gleichzeitige Schreibvorgänge (nicht geschützt)

```
/mybucket/important.doc
PUT#1 ---> OK
PUT#2 -------> OK
```

Situation B: Sequentielles Überschreiben abgeschlossen (vorbeugend)

```
/mybucket/important.doc
PUT#1 -------> PUT#2 ---X (denied)
```

Ähnliche Informationen

- "So verwalten StorageGRID ILM-Regeln Objekte"
- "Beispiele für Bucket-Richtlinien"
- "Beispiele für Gruppenrichtlinien"
- "Objekte mit ILM verwalten"
- "Verwenden eines Mandantenkontos"

Beispiele für Bucket-Richtlinien

Verwenden Sie die Beispiele in diesem Abschnitt, um StorageGRID Zugriffsrichtlinien für Buckets zu erstellen.

Bucket-Richtlinien geben die Zugriffsberechtigungen für den Bucket an, an den die Richtlinie angehängt ist. Sie konfigurieren eine Bucket-Richtlinie mithilfe der S3 PutBucketPolicy-API über eines dieser Tools:

- "Mietermanager" .
- AWS CLI mit diesem Befehl (siehe"Operationen an Buckets"):

```
> aws s3api put-bucket-policy --bucket examplebucket --policy
file://policy.json
```

Beispiel: Allen Lesezugriff auf einen Bucket gewähren

In diesem Beispiel darf jeder, auch anonyme Benutzer, Objekte im Bucket auflisten und GetObject-Operationen für alle Objekte im Bucket ausführen. Alle anderen Vorgänge werden abgelehnt. Beachten Sie, dass diese Richtlinie möglicherweise nicht besonders nützlich ist, da niemand außer dem Konto-Root über die Berechtigung zum Schreiben in den Bucket verfügt.

```
{
  "Statement": [
    {
      "Sid": "AllowEveryoneReadOnlyAccess",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [ "s3:GetObject", "s3:ListBucket" ],
      "Resource":
["arn:aws:s3:::examplebucket","arn:aws:s3:::examplebucket/*"]
    }
  ]
}
```

Beispiel: Erlauben Sie allen Benutzern eines Kontos den Vollzugriff und allen Benutzern eines anderen Kontos den Lesezugriff auf einen Bucket.

In diesem Beispiel erhält jeder in einem angegebenen Konto vollen Zugriff auf einen Bucket, während jeder in einem anderen angegebenen Konto nur den Bucket auflisten und GetObject-Operationen für Objekte im Bucket ausführen darf, beginnend mit dem `shared/` Objektschlüsselpräfix.

> In StorageGRID sind Objekte, die von einem Nicht-Eigentümerkonto (einschließlich anonymer Konten) erstellt wurden, Eigentum des Bucket-Eigentümerkontos. Für diese Objekte gilt die Bucket-Richtlinie.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "95390887230002558202"
      },
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "31181711887329436680"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::examplebucket/shared/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "31181711887329436680"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::examplebucket",
      "Condition": {
        "StringLike": {
          "s3:prefix": "shared/*"
        }
      }
    }
  ]
}
```

Beispiel: Erlauben Sie allen nur Lesezugriff auf einen Bucket und Vollzugriff für eine bestimmte Gruppe

In diesem Beispiel darf jeder, auch anonym, den Bucket auflisten und GetObject-Operationen für alle Objekte im Bucket durchführen, während nur Benutzer der Gruppe `Marketing` im angegebenen Konto wird der volle Zugriff gewährt.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-
group/Marketing"
      },
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": ["s3:ListBucket","s3:GetObject"],
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    }
  ]
}
```

Beispiel: Erlauben Sie jedem Lese- und Schreibzugriff auf einen Bucket, wenn sich der Client im IP-Bereich befindet

In diesem Beispiel darf jeder, auch anonyme Benutzer, den Bucket auflisten und beliebige Objektoperationen für alle Objekte im Bucket ausführen, vorausgesetzt, die Anforderungen stammen aus einem angegebenen IP-Bereich (54.240.143.0 bis 54.240.143.255, außer 54.240.143.188). Alle anderen Vorgänge werden abgelehnt und alle Anfragen außerhalb des IP-Bereichs werden abgelehnt.

```
{
  "Statement": [
    {
      "Sid": "AllowEveryoneReadWriteAccessIfInSourceIpRange",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [ "s3:*Object", "s3:ListBucket" ],
      "Resource":
["arn:aws:s3:::examplebucket","arn:aws:s3:::examplebucket/*"],
      "Condition": {
        "IpAddress": {"aws:SourceIp": "54.240.143.0/24"},
        "NotIpAddress": {"aws:SourceIp": "54.240.143.188"}
      }
    }
  ]
}
```

Beispiel: Vollzugriff auf einen Bucket ausschließlich durch einen angegebenen Verbundbenutzer zulassen

In diesem Beispiel erhält der Verbundbenutzer Alex vollen Zugriff auf die `examplebucket` Bucket und seine Objekte. Allen anderen Benutzern, einschließlich „root“, werden sämtliche Vorgänge ausdrücklich verweigert. Beachten Sie jedoch, dass „root“ niemals die Berechtigung zum Put/Get/DeleteBucketPolicy verweigert wird.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-user/Alex"
      },
      "Action": [
        "s3:*"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    },
    {
      "Effect": "Deny",
      "NotPrincipal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-user/Alex"
      },
      "Action": [
        "s3:*"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket",
        "arn:aws:s3:::examplebucket/*"
      ]
    }
  ]
}
```

Beispiel: PutOverwriteObject-Berechtigung

In diesem Beispiel `Deny` Die Wirkung von PutOverwriteObject und DeleteObject stellt sicher, dass niemand die Daten, benutzerdefinierten Metadaten und S3-Objektmarkierungen des Objekts überschreiben oder löschen kann.

```
{
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "s3:PutOverwriteObject",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
      ],
      "Resource": "arn:aws:s3:::wormbucket/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-
group/SomeGroup"

},
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::wormbucket"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::95390887230002558202:federated-
group/SomeGroup"

},
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::wormbucket/*"
    }
  ]
}
```

Beispiele für Gruppenrichtlinien

Verwenden Sie die Beispiele in diesem Abschnitt, um StorageGRID Zugriffsrichtlinien für Gruppen zu erstellen.

Gruppenrichtlinien geben die Zugriffsberechtigungen für die Gruppe an, der die Richtlinie zugeordnet ist. Es gibt keine `Principal` Element in der Richtlinie, da es implizit ist. Gruppenrichtlinien werden mithilfe des Tenant Managers oder der API konfiguriert.

Beispiel: Festlegen der Gruppenrichtlinie mit dem Mandanten-Manager

Wenn Sie im Mandanten-Manager eine Gruppe hinzufügen oder bearbeiten, können Sie eine Gruppenrichtlinie auswählen, um festzulegen, welche S3-Zugriffsberechtigungen die Mitglieder dieser Gruppe haben. Sehen "Erstellen von Gruppen für einen S3-Mandanten" .

- **Kein S3-Zugriff**: Standardoption. Benutzer in dieser Gruppe haben keinen Zugriff auf S3-Ressourcen, es sei denn, der Zugriff wird mit einer Bucket-Richtlinie gewährt. Wenn Sie diese Option auswählen, hat standardmäßig nur der Root-Benutzer Zugriff auf S3-Ressourcen.
- **Nur-Lesezugriff**: Benutzer in dieser Gruppe haben nur Lesezugriff auf S3-Ressourcen. Beispielsweise können Benutzer dieser Gruppe Objekte auflisten und Objektdaten, Metadaten und Tags lesen. Wenn Sie diese Option auswählen, wird die JSON-Zeichenfolge für eine schreibgeschützte Gruppenrichtlinie im Textfeld angezeigt. Sie können diese Zeichenfolge nicht bearbeiten.
- **Vollzugriff**: Benutzer in dieser Gruppe haben vollen Zugriff auf S3-Ressourcen, einschließlich Buckets. Wenn Sie diese Option auswählen, wird die JSON-Zeichenfolge für eine Gruppenrichtlinie mit vollem Zugriff im Textfeld angezeigt. Sie können diese Zeichenfolge nicht bearbeiten.
- **Ransomware-Minderung**: Diese Beispielrichtlinie gilt für alle Buckets dieses Mandanten. Benutzer in dieser Gruppe können allgemeine Aktionen ausführen, aber keine Objekte dauerhaft aus Buckets löschen, für die die Objektversionierung aktiviert ist.

  Tenant Manager-Benutzer mit der Berechtigung „Alle Buckets verwalten“ können diese Gruppenrichtlinie außer Kraft setzen. Beschränken Sie die Berechtigung „Alle Buckets verwalten“ auf vertrauenswürdige Benutzer und verwenden Sie, sofern verfügbar, die Multi-Faktor-Authentifizierung (MFA).

- **Benutzerdefiniert**: Benutzern in der Gruppe werden die Berechtigungen erteilt, die Sie im Textfeld angeben.

Beispiel: Gruppe vollen Zugriff auf alle Buckets gewähren

In diesem Beispiel wird allen Mitgliedern der Gruppe der vollständige Zugriff auf alle Buckets gewährt, die dem Mandantenkonto gehören, sofern die Bucket-Richtlinie diesen Zugriff nicht ausdrücklich verweigert.

```
{
  "Statement": [
    {
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::*"
    }
  ]
}
```

Beispiel: Gruppe schreibgeschützten Zugriff auf alle Buckets gewähren

In diesem Beispiel haben alle Mitglieder der Gruppe schreibgeschützten Zugriff auf S3-Ressourcen, sofern dies nicht ausdrücklich durch die Bucket-Richtlinie verweigert wird. Beispielsweise können Benutzer dieser Gruppe Objekte auflisten und Objektdaten, Metadaten und Tags lesen.

```
{
  "Statement": [
    {
      "Sid": "AllowGroupReadOnlyAccess",
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:ListBucketVersions",
        "s3:GetObject",
        "s3:GetObjectTagging",
        "s3:GetObjectVersion",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::*"
    }
  ]
}
```

Beispiel: Gruppenmitgliedern vollen Zugriff nur auf ihren „Ordner“ in einem Bucket gewähren

In diesem Beispiel dürfen Mitglieder der Gruppe nur ihren spezifischen Ordner (Schlüsselpräfix) im angegebenen Bucket auflisten und darauf zugreifen. Beachten Sie, dass bei der Festlegung des Datenschutzes dieser Ordner Zugriffsberechtigungen aus anderen Gruppenrichtlinien und der Bucket-Richtlinie berücksichtigt werden sollten.

```
{
  "Statement": [
    {
      "Sid": "AllowListBucketOfASpecificUserPrefix",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::department-bucket",
      "Condition": {
        "StringLike": {
          "s3:prefix": "${aws:username}/*"
        }
      }
    },
    {
      "Sid": "AllowUserSpecificActionsOnlyInTheSpecificUserPrefix",
      "Effect": "Allow",
      "Action": "s3:*Object",
      "Resource": "arn:aws:s3:::department-bucket/${aws:username}/*"
    }
  ]
}
```

In den Prüfprotokollen verfolgte S3-Operationen

Audit-Nachrichten werden von StorageGRID -Diensten generiert und in Textprotokolldateien gespeichert. Sie können die S3-spezifischen Prüfmeldungen im Prüfprotokoll überprüfen, um Details zu Bucket- und Objektvorgängen zu erhalten.

In den Audit-Protokollen verfolgte Bucket-Operationen

- Bucket erstellen
- Bucket löschen
- BucketTagging löschen
- Objekte löschen
- GetBucketTagging
- Kopfeimer
- ListObjects
- ListObjectVersions
- PUT Bucket-Konformität
- PutBucketTagging
- PutBucketVersioning

In den Überwachungsprotokollen verfolgte Objektvorgänge

- CompleteMultipartUpload
- Objekt kopieren
- Objekt löschen
- GetObject
- HeadObject
- PutObject
- RestoreObject
- Objekt auswählen
- UploadPart (wenn eine ILM-Regel eine ausgeglichene oder strikte Aufnahme verwendet)
- UploadPartCopy (wenn eine ILM-Regel eine ausgeglichene oder strikte Aufnahme verwendet)

Ähnliche Informationen

- "Zugriff auf die Überwachungsprotokolldatei"
- "Client schreibt Prüfmeldungen"
- "Client liest Audit-Nachrichten"