S3 REST-API verwenden

StorageGRID software

NetApp
July 23, 2026

Inhalt

S3 REST-API verwenden

Unterstützte Versionen und Updates der StorageGRID S3 REST API

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

Die Unterstützung der S3 REST API ermöglicht die Anbindung serviceorientierter Anwendungen, die für S3 Webdienste entwickelt wurden, an lokale Objektspeicher, die das StorageGRID System nutzen. Nur minimale Änderungen an der aktuellen Verwendung von S3 REST API-Aufrufen durch eine Clientanwendung sind erforderlich.

Unterstützte Versionen

StorageGRID unterstützt die folgenden spezifischen Versionen von STS, S3 und HTTP (Hypertext Transfer Protocol):

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

Aktualisierungen der S3 REST API Unterstützung

Freigabe	Kommentare
12,0	• Unterstützung für Cross-Origin Resource Sharing (CORS) für eine Managementoberfläche hinzugefügt, wodurch eine andere Domäne über Management-APIs auf Daten in StorageGRID zugreifen kann. "Mehr erfahren". • Unterstützung für STS AssumeRole und Sitzungsrichtlinien wurde hinzugefügt. Siehe "ein Beispiel für eine Sitzungsrichtlinie". AssumeRole kann unter Mandantengruppen eingerichtet werden.
11,9	• Unterstützung für vorab berechnete SHA-256-Prüfsummenwerte für die folgenden Anfragen und unterstützten Header hinzugefügt. Mit dieser Funktion lässt sich die Integrität hochgeladener Objekte überprüfen: ◦ CompleteMultipartUpload: `x-amz-checksum-sha256` ◦ CreateMultipartUpload: `x-amz-checksum-algorithm` ◦ GetObject: `x-amz-checksum-mode` ◦ HeadObject: `x-amz-checksum-mode` ◦ ListParts ◦ PutObject: `x-amz-checksum-sha256` ◦ UploadPart: `x-amz-checksum-sha256` • Die Möglichkeit für den Grid-Administrator, die Aufbewahrungs- und Compliance-Einstellungen auf Mandantenebene zu steuern, wurde hinzugefügt. Diese Einstellungen wirken sich auf die S3 Object Lock-Einstellungen aus. ◦ Standardmäßiger Aufbewahrungsmodus für Buckets und Aufbewahrungsmodus für Objekte: Governance oder Compliance, sofern vom Grid-Administrator zugelassen. ◦ Standardmäßige Aufbewahrungsdauer für Buckets und Aufbewahrungsdatum für Objekte: Muss kleiner oder gleich der vom Grid-Administrator festgelegten maximalen Aufbewahrungsdauer sein. • Verbesserte Unterstützung für `aws-chunked` Inhaltskodierung und Streaming `x-amz-content-sha256` Werte. Einschränkungen: ◦ Falls vorhanden, `chunk-signature` ist optional und wird nicht validiert ◦ Falls `x-amz-trailer`Inhalt vorhanden ist, wird er ignoriert
11,8	Die Namen der S3-Operationen wurden aktualisiert, um mit den in der "Amazon Web Services (AWS) Dokumentation: Amazon Simple Storage Service API Reference" verwendeten Namen übereinzustimmen.

Freigabe	Kommentare
11,7	• Hinzugefügt "Kurzübersicht: Unterstützte S3 API-Anfragen". • Unterstützung für die Verwendung des GOVERNANCE-Modus mit S3 Object Lock hinzugefügt. • Unterstützung für den StorageGRID-spezifischen `x-ntap-sg-cgr-replication-status` Response-Header für GET Object- und HEAD Object-Anfragen hinzugefügt. Dieser Header gibt den Replikationsstatus eines Objekts für die gridübergreifende Replikation an. • SelectObjectContent-Anfragen unterstützen jetzt Parquet-Objekte.
11,6	• Unterstützung für die Verwendung des `partNumber` Anfrageparameters in GET Object- und HEAD Object-Anfragen hinzugefügt. • Unterstützung für einen Standard-Aufbewahrungsmodus und eine Standard-Aufbewahrungsdauer auf Bucket-Ebene für S3 Object Lock. • Unterstützung für den `s3:object-lock-remaining-retention-days`Richtlinienbedingungsschlüssel wurde hinzugefügt, um den Bereich der zulässigen Aufbewahrungsfristen für Ihre Objekte festzulegen. • Die maximal empfohlene Größe für eine einzelne PUT Object-Operation wurde auf 5 GiB (5.368.709.120 Byte) geändert. Bei Objekten, die größer als 5 GiB sind, sollte stattdessen ein Multipart-Upload verwendet werden.
11,5	• Unterstützung für die Verwaltung der Bucket-Verschlüsselung wurde hinzugefügt. • Unterstützung für S3 Object Lock und veraltete Compliance-Anforderungen hinzugefügt. • Unterstützung für die Verwendung von DELETE Multiple Objects auf versionierten Buckets wurde hinzugefügt. • Der `Content-MD5`Anfrageheader wird jetzt korrekt unterstützt.
11,4	• Unterstützung für DELETE Bucket tagging, GET Bucket tagging und PUT Bucket tagging wurde hinzugefügt. Kostenallokations-Tags werden nicht unterstützt. • Für Buckets, die in StorageGRID 11.4 erstellt wurden, ist es nicht mehr erforderlich, Objektschlüsselnamen einzuschränken, um Best Practices für die Leistung einzuhalten. • Unterstützung für Bucket-Benachrichtigungen für den `s3:ObjectRestore:Post` Ereignistyp wurde hinzugefügt. • AWS-Größenbeschränkungen für Multipart-Teile werden jetzt durchgesetzt. Jeder Teil eines Multipart-Uploads muss zwischen 5 MiB und 5 GiB liegen. Der letzte Teil kann kleiner als 5 MiB sein. • Unterstützung für TLS 1.3 hinzugefügt

Freigabe	Kommentare
11,3	• Unterstützung für die serverseitige Verschlüsselung von Objektdaten mit vom Kunden bereitgestellten Schlüsseln (SSE-C) wurde hinzugefügt. • Unterstützung für DELETE-, GET- und PUT-Bucket-Lifecycle-Operationen (nur Expiration-Aktion) und für den `x-amz-expiration`Response-Header hinzugefügt. • PUT Object, PUT Object - Copy und Multipart Upload wurden aktualisiert, um die Auswirkungen von ILM-Regeln zu beschreiben, die synchrones Placement beim Ingest verwenden. • TLS 1.1 Verschlüsselungen werden nicht mehr unterstützt.
11,2	Unterstützung für die Wiederherstellung von Objekten per POST-Anfrage für die Verwendung mit Cloud Storage Pools hinzugefügt. Unterstützung für die Verwendung der AWS-Syntax für ARN, Richtlinienbedingungsschlüssel und Richtlinienvariablen in Gruppen- und Bucket-Richtlinien hinzugefügt. Bestehende Gruppen- und Bucket-Richtlinien, die die StorageGRID Syntax verwenden, werden weiterhin unterstützt. **Hinweis:** Die Verwendung von ARN/URN in anderen Konfigurations-JSON/XML, einschließlich derjenigen, die in benutzerdefinierten StorageGRID Features verwendet werden, 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 wurde hinzugefügt.
11,0	Unterstützung für die Konfiguration von Plattformdiensten (CloudMirror Replikation, Benachrichtigungen und Elasticsearch-Suchintegration) für Buckets wurde hinzugefügt. Ebenfalls wurde Unterstützung für Standortbeschränkungen bei Objekt-Tags für Buckets sowie für die Available-Konsistenz hinzugefügt.
10,4	Unterstützung für ILM-Scanning-Änderungen an der Versionierung, Aktualisierungen der Seite „Endpoint Domain Names“, Bedingungen und Variablen in Richtlinien, Richtlinienbeispiele und die Berechtigung PutOverwriteObject wurde hinzugefügt.
10,3	Unterstützung für Versionierung hinzugefügt.
10,2	Unterstützung für Gruppen- und Bucket-Zugriffsrichtlinien sowie für mehrteilige Kopien (Upload Part - Copy) hinzugefügt.
10,1	Unterstützung für Multipart-Upload, virtuelle Hosted-Anfragen und v4 Authentifizierung.
10,0	Erste Unterstützung der S3 REST API durch das StorageGRID System. Die aktuell unterstützte Version der *Simple Storage Service API Reference* ist 2006-03-01.

Unterstützte S3-API-Anfragen in StorageGRID

Diese Seite fasst zusammen, wie StorageGRID die Amazon Simple Storage Service (S3) APIs unterstützt.

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

> Die AWS-Dokumentation zu den einzelnen Operationen ist über den Link in der Überschrift abrufbar.

Gängige URI-Abfrageparameter und Anfrageheader

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

- `versionId` (wie für Objektoperationen erforderlich)

Sofern nicht anders angegeben, werden die folgenden gängigen Anfrage-Header unterstützt:

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

Verwandte Informationen

- "Details zur Implementierung der S3 REST API"
- "Amazon Simple Storage Service API Reference: Allgemeine Anfrageheader"

"AbortMultipartUpload"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage sowie diesen zusätzlichen URI-Abfrageparameter:

- `uploadId`

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen für mehrteilige Uploads"

"CompleteMultipartUpload"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage sowie diesen zusätzlichen URI-Abfrageparameter:

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

XML-Tags im Anfragetext

StorageGRID unterstützt diese XML-Tags im Anfragetext:

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

StorageGRID Dokumentation

"CompleteMultipartUpload"

"CopyObject"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage sowie die folgenden 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>`

Anfragekörper

Keine

StorageGRID Dokumentation

"CopyObject"

"CreateBucket"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage sowie die folgenden zusätzlichen Header:

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

Anfragekörper

StorageGRID unterstützt alle Anfragekörperparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST API definiert sind.

StorageGRID Dokumentation

"Operationen an Buckets"

"CreateMultipartUpload"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage sowie die folgenden 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>`

Anfragekörper

Keine

StorageGRID Dokumentation

"CreateMultipartUpload"

"DeleteBucket"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

StorageGRID Dokumentation

"Operationen an Buckets"

"DeleteBucketCors"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"DeleteBucketEncryption"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"DeleteBucketLifecycle"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

- "Operationen an Buckets"
- "S3 Lifecycle-Konfiguration erstellen"

"DeleteBucketPolicy"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"DeleteBucketReplication"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"DeleteBucketTagging"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"DeleteObject"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage sowie diesen zusätzlichen Anfrageheader:

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

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Objekten"

"DeleteObjects"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage sowie diesen zusätzlichen Anfrageheader:

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

Anfragekörper

StorageGRID unterstützt alle Anfragekörperparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST API definiert sind.

StorageGRID Dokumentation

"Operationen an Objekten"

"DeleteObjectTagging"

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Objekten"

"GetBucketAcl"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"GetBucketCors"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"GetBucketEncryption"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"GetBucketLifecycleConfiguration"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

- "Operationen an Buckets"
- "S3 Lifecycle-Konfiguration erstellen"

"GetBucketLocation"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"GetBucketNotificationConfiguration"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"GetBucketPolicy"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"GetBucketReplication"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"GetBucketTagging"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"GetBucketVersioning"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"GetObject"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage erforderlichen Parameter sowie die folgenden 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 Anfrage-Header:

- `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`

Anfragekörper

Keine

StorageGRID Dokumentation

"GetObject"

"GetObjectAcl"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Objekten"

"GetObjectLegalHold"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"S3 REST API zur Konfiguration von S3 Object Lock verwenden"

"GetObjectLockConfiguration"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"S3 REST API zur Konfiguration von S3 Object Lock verwenden"

"GetObjectRetention"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"S3 REST API zur Konfiguration von S3 Object Lock verwenden"

"GetObjectTagging"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Objekten"

"HeadBucket"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"HeadObject"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage sowie die folgenden 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`

Anfragekörper

Keine

StorageGRID Dokumentation

"HeadObject"

"ListBuckets"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Keine

StorageGRID Dokumentation

Operationen am Service › ListBuckets

"ListMultipartUploads"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage sowie die folgenden zusätzlichen Parameter:

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

Anfragekörper

Keine

StorageGRID Dokumentation

"ListMultipartUploads"

"ListObjects"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage sowie die folgenden zusätzlichen Parameter:

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

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"ListObjectsV2"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage sowie die folgenden zusätzlichen Parameter:

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

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"ListObjectVersions"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage sowie die folgenden zusätzlichen Parameter:

- `delimiter`

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

Anfragekörper

Keine

StorageGRID Dokumentation

"Operationen an Buckets"

"ListParts"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage sowie die folgenden zusätzlichen Parameter:

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

Anfragekörper

Keine

StorageGRID Dokumentation

"ListMultipartUploads"

"PutBucketCors"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

StorageGRID unterstützt alle Anfragekörperparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST API definiert sind.

StorageGRID Dokumentation

"Operationen an Buckets"

"PutBucketEncryption"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

XML-Tags im Anfragetext

StorageGRID unterstützt diese XML-Tags im Anfragetext:

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

StorageGRID Dokumentation

"Operationen an Buckets"

"PutBucketLifecycleConfiguration"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

XML-Tags im Anfragetext

StorageGRID unterstützt diese XML-Tags im Anfragetext:

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

StorageGRID Dokumentation

- "Operationen an Buckets"
- "S3 Lifecycle-Konfiguration erstellen"

"PutBucketNotificationConfiguration"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

XML-Tags im Anfragetext

StorageGRID unterstützt diese XML-Tags im Anfragetext:

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

StorageGRID Dokumentation

"Operationen an Buckets"

"PutBucketPolicy"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

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

"PutBucketReplication"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

XML-Tags im Anfragetext

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

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

StorageGRID Dokumentation

"Operationen an Buckets"

"PutBucketTagging"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

StorageGRID unterstützt alle Anfragekörperparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST API definiert sind.

StorageGRID Dokumentation

"Operationen an Buckets"

"PutBucketVersioning"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Parameter des Anfragetextes

StorageGRID unterstützt die folgenden Anfragekörperparameter:

- `VersioningConfiguration`
- `Status`

StorageGRID Dokumentation

"Operationen an Buckets"

"PutObject"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage sowie die folgenden 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>`

Anfragekörper

- Binärdaten des Objekts

StorageGRID Dokumentation

"PutObject"

"PutObjectLegalHold"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

StorageGRID unterstützt alle Anfragekörperparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST API definiert sind.

StorageGRID Dokumentation

"S3 REST API zur Konfiguration von S3 Object Lock verwenden"

"PutObjectLockConfiguration"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

StorageGRID unterstützt alle Anfragekörperparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST API definiert sind.

StorageGRID Dokumentation

"S3 REST API zur Konfiguration von S3 Object Lock verwenden"

"PutObjectRetention"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage sowie diesen zusätzlichen Header:

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

Anfragekörper

StorageGRID unterstützt alle Anfragekörperparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST API definiert sind.

StorageGRID Dokumentation

"S3 REST API zur Konfiguration von S3 Object Lock verwenden"

"PutObjectTagging"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

StorageGRID unterstützt alle Anfragekörperparameter, die zum Zeitpunkt der Implementierung von der Amazon S3 REST API definiert sind.

StorageGRID Dokumentation

"Operationen an Objekten"

"RestoreObject"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Details zu den unterstützten Textfeldern finden Sie unter "RestoreObject".

"SelectObjectContent"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage.

Anfragekörper

Einzelheiten zu den unterstützten Body-Feldern sind im Folgenden aufgeführt:

- "S3 Select nutzen"
- "SelectObjectContent"

"UploadPart"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage erforderlichen Parameter sowie die folgenden zusätzlichen URI-Abfrageparameter:

- `partNumber`
- `uploadId`

Und diese zusätzlichen Anfrage-Header:

- `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`

Anfragekörper

- Binärdaten des Teils

StorageGRID Dokumentation

"UploadPart"

"UploadPartCopy"

URI-Abfrageparameter und Anfrageheader

StorageGRID unterstützt alle allgemeine Parameter und Header für diese Anfrage erforderlichen Parameter sowie die folgenden zusätzlichen URI-Abfrageparameter:

- `partNumber`
- `uploadId`

Und diese zusätzlichen Anfrage-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-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`

Anfragekörper

Keine

StorageGRID Dokumentation

"UploadPartCopy"

Testen Sie die StorageGRID S3 REST API-Konfiguration

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

Bevor Sie beginnen

- Sie haben die AWS CLI von "aws.amazon.com/cli" heruntergeladen und installiert.
- Optional haben Sie "Ein Load Balancer-Endpunkt wurde erstellt.". Andernfalls ist die IP-Adresse des Storage Node, zu dem die Verbindung hergestellt werden soll, sowie die zu verwendende Portnummer bekannt. Siehe "IP-Adressen und Ports für Clientverbindungen".
- Du hast "Ein S3-Mandantenkonto wurde erstellt".
- Sie haben sich beim Mandanten angemeldet und "hat einen Zugriffsschlüssel erstellt".

Einzelheiten zu diesen Schritten finden sich unter "Clientverbindungen konfigurieren".

Schritte

1. Die AWS CLI-Einstellungen werden so konfiguriert, dass das im StorageGRID System erstellte Konto verwendet wird:
   a. Konfigurationsmodus aktivieren: `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 standardmäßig zu verwendende Region ein. Zum Beispiel `us-east-1`.
   e. Geben Sie das Standardausgabeformat ein oder drücken Sie **Enter**, um JSON auszuwählen.
2. Einen Bucket erstellen.

   In diesem Beispiel wird davon ausgegangen, dass ein Load-Balancer-Endpunkt so konfiguriert wurde, dass die IP-Adresse 10.96.101.17 und der Port 10443 verwendet werden.

   ```
   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. Ein Objekt hochladen.

   ```
   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 wird, wird ein Etag zurückgegeben, der ein Hash der Objektdaten ist.

4. Den Inhalt des Buckets auflisten, um zu überprüfen, dass das Objekt hochgeladen wurde.

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

5. Das Objekt löschen.

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

6. Den Bucket löschen.

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

Wie StorageGRID die S3 REST API implementiert

Wie StorageGRID S3 REST API widersprüchliche Clientanfragen löst

Widersprüchliche Clientanfragen, wie zum Beispiel das Schreiben zweier Clients auf denselben Schlüssel, werden nach dem Prinzip „latest-wins“ gelöst.

Der Zeitpunkt für die Auswertung „latest-wins“ richtet sich danach, wann das StorageGRID System eine bestimmte Anfrage abschließt, und nicht danach, wann S3 Clients eine Operation starten.

Wie StorageGRID S3 REST API Verfügbarkeit und Konsistenz in Einklang bringt

Konsistenz sorgt für ein Gleichgewicht zwischen der Verfügbarkeit der Objekte und der Konsistenz dieser Objekte über verschiedene Storage Nodes und Standorte hinweg. Die Konsistenz kann entsprechend den Anforderungen Ihrer Anwendung geändert werden.

Standardmäßig garantiert StorageGRID Lesekonsistenz nach dem Schreiben für neu erstellte Objekte. Jeder GET nach einem erfolgreich abgeschlossenen PUT kann die neu geschriebenen Daten lesen. Das Überschreiben bestehender Objekte, Metadatenaktualisierungen und Löschvorgänge sind schließlich konsistent.

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

- Eine Konsistenz für jeder Bucket angeben.
- Eine Konsistenz für jede API-Operation angeben.
- Die standardmäßige netzweite Konsistenz kann durch Ausführen einer der folgenden Aufgaben geändert werden:

- Im Grid Manager zu **Konfiguration** > **System** > **Speichereinstellungen** > **Standardmäßige Bucket-Konsistenz** navigieren.
- .

> Eine Änderung der netzweiten Konsistenz gilt nur für Buckets, die nach der Änderung der Einstellung erstellt wurden. Die Details einer Änderung sind im Revisionsprotokoll unter `/var/local/log` zu finden (Suche nach **consistencyLevel**).

Konsistenzwerte

Die Konsistenz beeinflusst, wie die Metadaten, die StorageGRID zur Objektverfolgung verwendet, zwischen den Knoten verteilt werden. Die Konsistenz beeinflusst die Verfügbarkeit von Objekten für Clientanfragen.

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

- **Alle**: Alle Knoten erhalten die Objektmetadaten sofort oder die Anfrage schlägt fehl.
- **Stark-global**: Gewährleistet Lese-nach-Schreib-Konsistenz für alle Clientanfragen an allen Standorten. Bei konfigurierter Quorum-Semantik gelten folgende Verhaltensweisen:
  - Ermöglicht die Standortausfalltoleranz für Clientanfragen bei Grids mit drei oder mehr Standorten. Grids mit nur zwei Standorten bieten keine Standortausfalltoleranz.
  - Die folgenden S3-Operationen sind bei einem Standortausfall nicht erfolgreich:
    - DeleteBucketEncryption
    - PutBucketBranch
    - PutBucketEncryption
    - PutBucketVersioning
    - PutObjectLegalHold
    - PutObjectLockConfiguration
    - PutObjectRetention

  Falls erforderlich kann "StorageGRID Quorumsemantik für starke globale Konsistenz konfigurieren" verwendet werden.
- **Strong-site**: Objektmetadaten werden sofort an andere Knoten am Standort verteilt. Garantiert Lese-nach-Schreib-Konsistenz für alle Clientanfragen innerhalb eines Standorts.
- **Read-after-new-write**: Gewährleistet Lesekonsistenz nach dem Schreiben für neue Objekte und letztendliche Konsistenz für Objektaktualisierungen. Bietet hohe Verfügbarkeit und Garantien für den Datenschutz. Empfohlen für die meisten Fälle.
- **Verfügbar**: Gewährleistet letztendliche Konsistenz sowohl für neue Objekte als auch für Objektaktualisierungen. Für S3-Buckets nur bei Bedarf verwenden (zum Beispiel für einen Bucket, der Log-Werte enthält, die selten gelesen werden, oder für HEAD- oder GET-Operationen auf Schlüsseln, die nicht existieren). Nicht unterstützt für S3 FabricPool Buckets.

Die Konsistenzoptionen „Read-after-new-write“ und „Available“ verwenden

Wenn eine HEAD- oder GET-Operation die Konsistenz "Read-after-new-write" verwendet, führt StorageGRID die Suche in mehreren Schritten durch, wie folgt:

- Zuerst wird das Objekt mit niedriger Konsistenz gesucht.
- Schlägt diese Suche fehl, wird die Suche beim nächsten Konsistenzwert wiederholt, bis eine Konsistenz erreicht ist, die dem Verhalten für strong-global entspricht.

Wenn eine HEAD- oder GET-Operation die Konsistenz „Read-after-new-write“ verwendet, das Objekt aber nicht existiert, erreicht die Objektsuche immer eine Konsistenz, die dem Verhalten bei strong-global entspricht. Da diese Konsistenz voraussetzt, dass an jedem Standort mehrere Kopien der Objektmetadaten verfügbar sind, kann eine hohe Anzahl von 500 Internal Server Errors auftreten, wenn zwei oder mehr Storage Nodes am selben Standort nicht verfügbar sind.

Sofern Sie keine Konsistenzgarantien wie bei Amazon S3 benötigen, können Sie diese Fehler bei HEAD- und GET-Operationen vermeiden, indem Sie die Konsistenz auf „Verfügbar“ setzen. Wenn eine HEAD- oder GET-Operation die Konsistenz „Verfügbar“ verwendet, bietet StorageGRID lediglich letztendliche Konsistenz. Es wird keine fehlgeschlagene Operation mit erhöhter Konsistenz wiederholt, sodass keine mehreren Kopien der Objektmetadaten erforderlich sind.

Konsistenz für API-Operationen angeben

Um die Konsistenz für eine einzelne API-Operation festzulegen, müssen die Konsistenzwerte für die Operation unterstützt werden und die Konsistenz im Anfrageheader angegeben werden. In diesem Beispiel wird die Konsistenz für eine GetObject-Operation auf „Strong-site“ gesetzt.

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

Für die PutObject- und GetObject-Operationen muss die gleiche Konsistenz verwendet werden.

Konsistenz für Bucket angeben

Um die Konsistenz für einen Bucket festzulegen, kann die StorageGRID "PUT Bucket Konsistenz" Anfrage verwendet werden. Alternativ ist dies auch "die Konsistenz eines Buckets ändern" aus dem Tenant Manager möglich.

Beim Festlegen der Konsistenz für einen Bucket ist Folgendes zu beachten:

- Durch die Festlegung 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 durchgeführt werden. Dies wirkt sich nicht auf Operationen am Bucket selbst aus.
- Die Konsistenz für eine einzelne API-Operation hat Vorrang vor der Konsistenz für den Bucket.
- Buckets sollten im Allgemeinen die Standardkonsistenz „Read-after-new-write“ verwenden. Falls Anfragen nicht korrekt funktionieren, sollte nach Möglichkeit das Verhalten des Anwendungsclients geändert werden. Alternativ kann der Client so konfiguriert werden, dass er die Konsistenz für jede API-Anfrage einzeln festlegt. Die Konsistenz auf Bucket-Ebene sollte nur als letzte Möglichkeit festgelegt werden.

Wie Konsistenz- und ILM-Regeln interagieren und den Datenschutz beeinflussen

Sowohl die Wahl der Konsistenz als auch die ILM-Regel beeinflussen, wie Objekte geschützt werden. Diese

Einstellungen können miteinander interagieren.

Die beim Speichern eines Objekts verwendete Konsistenz beeinflusst beispielsweise die anfängliche Platzierung der Objektmetadaten, während das für die ILM-Regel ausgewählte Aufnahmeverhalten die anfängliche Platzierung von Objektkopien bestimmt. Da StorageGRID sowohl auf die Metadaten als auch auf die Daten eines Objekts zugreifen muss, um Clientanfragen zu erfüllen, kann die Auswahl übereinstimmender Schutzstufen für Konsistenz und Aufnahmeverhalten einen besseren anfänglichen Datenschutz und vorhersehbarere Systemreaktionen bieten.

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

Dual Commit
: StorageGRID erstellt umgehend temporäre Kopien des Objekts und gibt dem Client eine Erfolgsmeldung zurück. Kopien, die in der ILM-Regel angegeben sind, werden nach Möglichkeit erstellt.

Strikt
: Alle in der ILM-Regel angegebenen Kopien müssen erstellt werden, bevor dem Client eine Erfolgsmeldung zurückgegeben wird.

Ausgewogen
: StorageGRID versucht, beim Datenimport alle in der ILM-Regel festgelegten Kopien zu erstellen. Ist dies nicht möglich, werden Zwischenkopien erstellt und dem Client eine Erfolgsmeldung zurückgegeben. Die in der ILM-Regel festgelegten Kopien werden, sofern möglich, erstellt.

Beispiel dafür, wie Konsistenz- und ILM-Regel miteinander interagieren können

Angenommen, Sie verfügen über ein Drei-Standort-Grid mit der folgenden ILM-Regel und der folgenden Konsistenz:

- **ILM-Regel**: Drei Objektkopien werden erstellt, eine am lokalen Standort und je eine an jedem Remote-Standort. Das strikte Aufnahmeverhalten wird verwendet.
- **Konsistenz**: Stark-global (Objektmetadaten werden sofort an mehrere Standorte verteilt).

Wenn ein Client ein Objekt im Grid speichert, erstellt StorageGRID alle drei Objektkopien und verteilt Metadaten an mehrere Standorte, bevor dem Client eine Erfolgsmeldung zurückgegeben wird.

Das Objekt ist zum Zeitpunkt der erfolgreichen Datenerfassung vollständig vor Verlust geschützt. Wenn beispielsweise der lokale Standort kurz nach der Datenerfassung ausfällt, sind Kopien sowohl der Objektdaten als auch der Objektmetadaten weiterhin an den Remote-Standorten vorhanden. Das Objekt ist von den anderen Standorten vollständig abrufbar.

Wenn stattdessen dieselbe ILM-Regel und die starke Standortkonsistenz verwendet werden, kann der Client eine Erfolgsmeldung erhalten, nachdem die Objektdaten auf die Remote-Standorte repliziert wurden, aber bevor die Objektmetadaten dort verteilt sind. In diesem Fall entspricht der Schutzgrad der Objektmetadaten nicht dem Schutzgrad der Objektdaten. Geht der lokale Standort kurz nach der Aufnahme verloren, 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, falls Sie Unterstützung benötigen.

Wie StorageGRID die Objektversionierung nutzt

Sie können den Versionsstatus eines Buckets festlegen, wenn Sie mehrere Versionen

jedes Objekts aufbewahren möchten. Die Aktivierung der Versionierung für einen Bucket kann vor versehentlichem Löschen von Objekten schützen und ermöglicht das Abrufen und Wiederherstellen früherer Versionen eines Objekts.

Das StorageGRID System implementiert Versionierung mit Unterstützung für die meisten Funktionen, jedoch 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-Lifecycle-Konfiguration kombiniert werden. Die Versionierung muss für jeden Bucket explizit aktiviert werden. Wenn die Versionierung für einen Bucket aktiviert ist, erhält jedes dem Bucket hinzugefügte Objekt eine Versions-ID, die vom StorageGRID System generiert wird.

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

> Die Versionsverwaltung 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 durchsucht kontinuierlich alle Objekte und bewertet sie anhand der aktuellen ILM-Richtlinie neu. Alle Änderungen an ILM-Richtlinien werden auf alle zuvor erfassten Objekte angewendet. Dies schließt zuvor erfasste Versionen ein, wenn die Versionierung aktiviert ist. Die ILM-Überprüfung wendet neue ILM-Änderungen auf zuvor erfasste Objekte an.

Für S3-Objekte in versioning-fähigen Buckets ermöglicht die Versionierungsunterstützung die Erstellung von ILM-Regeln, die „Nicht-aktuelle Zeit“ als Referenzzeit verwenden (bei der Frage „Diese Regel nur auf ältere Objektversionen anwenden?“ in "Schritt 1 des Assistenten zum Erstellen einer ILM Regel" **Ja** auswählen). Wenn ein Objekt aktualisiert wird, werden die vorherigen Versionen zu nicht-aktuellen Versionen. Mit einem Filter für „Nicht-aktuelle Zeit“ lassen sich Richtlinien erstellen, die den Speicherbedarf vorheriger Objektversionen reduzieren.

> Wenn Sie eine neue Version eines Objekts mit einer Multipart-Upload-Operation hochladen, entspricht der nicht aktuelle Zeitpunkt der ursprünglichen Version dem Zeitpunkt, zu dem der Multipart-Upload für die neue Version erstellt wurde, nicht dem Zeitpunkt, zu dem der Multipart-Upload abgeschlossen wurde. In einigen Fällen kann der nicht aktuelle Zeitpunkt der ursprünglichen Version Stunden oder Tage vor dem Zeitpunkt der aktuellen Version liegen.

Verwandte Informationen

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

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

Wenn die globale S3 Object Lock-Einstellung für Ihr StorageGRID System aktiviert ist, können Buckets mit aktivierter S3 Object Lock-Funktion erstellt werden. Für jeden Bucket kann eine Standardaufbewahrung oder für jede Objektversion eine Aufbewahrungseinstellung angegeben werden.

Aktivierung der S3 Object Lock für einen Bucket

Wenn die globale S3 Object Lock-Einstellung für Ihr StorageGRID System aktiviert ist, kann die S3 Object Lock-Option beim Erstellen jedes Buckets optional aktiviert werden.

S3 Object Lock ist eine permanente Einstellung, die nur beim Erstellen eines Buckets aktiviert werden kann. S3 Object Lock kann nach der Erstellung eines Buckets nicht hinzugefügt oder deaktiviert werden.

Um die S3-Objektsperre für einen Bucket zu aktivieren, kann eine der folgenden Methoden verwendet werden:

- Der Bucket wird mithilfe des Tenant Manager erstellt. Siehe "S3 Bucket erstellen".
- Der Bucket wird mit einer CreateBucket-Anfrage und dem `x-amz-bucket-object-lock-enabled` Anfrageheader erstellt. Siehe "Operationen an Buckets".

S3 Object Lock erfordert die Bucket-Versionierung, die beim Erstellen des Buckets automatisch aktiviert wird. Die Versionierung des Buckets kann nicht deaktiviert werden. Siehe "Objektversionierung".

Standardmäßige Aufbewahrungseinstellungen für einen Bucket

Wenn S3 Object Lock für einen Bucket aktiviert ist, kann optional die Standardaufbewahrung für den Bucket aktiviert und ein Standardaufbewahrungsmodus sowie eine Standardaufbewahrungsdauer festgelegt werden.

Standard-Aufbewahrungsmodus

- 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 nach Erreichen dieses Datums entfernt werden.
- Im GOVERNANCE Modus:
  - Benutzer mit der `s3:BypassGovernanceRetention` Berechtigung können den `x-amz-bypass-governance-retention: true` Anfrageheader verwenden, um die Aufbewahrungseinstellungen zu umgehen.
  - Diese Benutzer können eine Objektversion löschen, bevor deren Aufbewahrungsdatum erreicht ist.
  - Diese Benutzer können das Aufbewahrungsdatum eines Objekts erhöhen, verringern oder entfernen.

Standardmäßige Aufbewahrungsdauer

Für jeden Bucket kann eine Standard-Aufbewahrungsfrist in Jahren oder Tagen festgelegt werden.

Festlegen der Standardaufbewahrungsdauer für einen Bucket

Um die Standardaufbewahrungsdauer für einen Bucket festzulegen, kann eine der folgenden Methoden verwendet werden:

- Bucket-Einstellungen können im Tenant Manager verwaltet werden. Siehe "Einen S3-Bucket erstellen" und "Standardaufbewahrung für S3 Object Lock aktualisieren".
- Eine PutObjectLockConfiguration-Anfrage für den Bucket ausgeben, um den Standardmodus und die Standardanzahl von Tagen oder Jahren festzulegen.

PutObjectLockConfiguration

Die PutObjectLockConfiguration-Anfrage ermöglicht das Festlegen und Ändern des Standardaufbewahrungsmodus und der Standardaufbewahrungsdauer für einen Bucket mit aktiviertem S3 Object Lock. Auch das Entfernen zuvor konfigurierter Standardaufbewahrungseinstellungen ist möglich.

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` nicht angegeben sind. Die Standardaufbewahrungsdauer wird zur Berechnung des Aufbewahrungsdatums verwendet, wenn `x-amz-object-lock-retain-until-date` nicht angegeben ist.

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

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

Der `Content-MD5`Anfrageheader muss in der PUT-Anfrage angegeben werden.

Anfragebeispiel

Dieses Beispiel aktiviert die S3 Object Lock für einen Bucket und legt den Standard-Aufbewahrungsmodus auf COMPLIANCE und die Standard-Aufbewahrungsdauer 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 wird die Standardaufbewahrungsdauer für einen Bucket ermittelt

${post_edited_translations.segment}

- Der Bucket ist im Tenant Manager sichtbar. Weitere Informationen unter

"${post_edited_translations.segment}".

- Eine GetObjectLockConfiguration-Anfrage ausgeben.

GetObjectLockConfiguration

Die GetObjectLockConfiguration-Anfrage ermöglicht die Feststellung, ob S3 Object Lock für einen Bucket aktiviert ist und, falls dies der Fall ist, ob ein Standard-Aufbewahrungsmodus und eine Standard-Aufbewahrungsdauer für den Bucket konfiguriert sind.

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

Sie müssen über die `s3:GetBucketObjectLockConfiguration`Berechtigung verfügen oder Kontoinhaber (root) sein, um diesen Vorgang abzuschließen.

Anfragebeispiel

```
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>
```

Festlegen von Aufbewahrungseinstellungen für ein Objekt

${post_edited_translations.segment}

Die Aufbewahrungseinstellungen auf Objektebene werden über die S3 REST API festgelegt. Die Aufbewahrungseinstellungen für ein Objekt überschreiben alle Standard-Aufbewahrungseinstellungen für den Bucket.

Für jedes Objekt lassen sich die folgenden Einstellungen angeben:

- ${post_edited_translations.segment}
- **Retain-until-date**: Ein Datum, das angibt, wie lange die Objektversion von StorageGRID aufbewahrt werden muss.
  - Im COMPLIANCE Modus kann das Objekt abgerufen werden, aber es kann nicht geändert oder gelöscht werden, wenn das Aufbewahrungsdatum in der Zukunft liegt. Das Aufbewahrungsdatum kann verlängert werden, aber dieses Datum kann nicht verkürzt oder entfernt werden.
  - Im GOVERNANCE-Modus können Benutzer mit einer speziellen Berechtigung die Einstellung für das Aufbewahrungsdatum umgehen. Sie können eine Objektversion löschen, bevor deren Aufbewahrungszeitraum abgelaufen ist. Sie können das Aufbewahrungsdatum auch heraufsetzen, herabsetzen oder sogar entfernen.
- **Legal Hold**: Das Anwenden eines Legal Hold auf eine Objektversion sperrt dieses Objekt sofort. Beispielsweise kann es erforderlich sein, einen Legal Hold für ein Objekt einzurichten, das im Zusammenhang mit einer Untersuchung oder einem Rechtsstreit steht. Ein Legal Hold hat kein Ablaufdatum, sondern bleibt bestehen, bis er explizit entfernt wird.

  Die Legal-Hold-Einstellung für ein Objekt ist unabhängig vom Aufbewahrungsmodus und dem Retain-until-Date. Wenn eine Objektversion unter einem Legal Hold steht, kann niemand diese Version löschen.

Um beim Hinzufügen einer Objektversion zu einem Bucket S3 Object Lock Einstellungen anzugeben, muss eine "PutObject"-, "CopyObject"- oder "CreateMultipartUpload"-Anforderung gesendet werden.

${post_edited_translations.segment}

- `x-amz-object-lock-mode`, was COMPLIANCE oder GOVERNANCE sein kann (Groß-/Kleinschreibung).

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

- `x-amz-object-lock-retain-until-date`
  - Der Wert für die Aufbewahrungsfrist muss im Format `2020-08-10T21:46:00Z` angegeben werden. Sekundenbruchteile sind zulässig, jedoch werden nur 3 Dezimalstellen beibehalten (Millisekundenpräzision). Andere ISO 8601 Formate sind nicht zulässig.
  - Das Aufbewahrungsdatum muss in der Zukunft liegen.
- `x-amz-object-lock-legal-hold`

  Wenn die rechtliche Aufbewahrung aktiviert ist (Groß-/Kleinschreibung beachten), wird das Objekt unter eine rechtliche Aufbewahrung gestellt. Wenn die rechtliche Aufbewahrung deaktiviert ist, wird keine rechtliche Aufbewahrung angewendet. Jeder andere Wert führt zu einem 400 Bad Request (InvalidArgument) Fehler.

Wenn Sie einen dieser Anfrage-Header verwenden, gelten folgende Einschränkungen:

- Der `Content-MD5` Anfrageheader ist erforderlich, wenn ein `x-amz-object-lock-*` Anfrageheader in der PutObject Anfrage vorhanden ist. `Content-MD5` ist nicht erforderlich für CopyObject oder CreateMultipartUpload.
- Wenn für den Bucket die S3 Object Lock-Funktion nicht aktiviert ist und ein `x-amz-object-lock-*` Request-Header vorhanden ist, wird ein 400 Bad Request (InvalidRequest)-Fehler zurückgegeben.
- Die PutObject-Anfrage unterstützt die Verwendung von `x-amz-storage-class: REDUCED_REDUNDANCY`, um das Verhalten von AWS nachzubilden. Wenn jedoch ein Objekt in einen Bucket mit aktiviertem S3 Object Lock aufgenommen wird, führt StorageGRID immer eine Dual-Commit-Aufnahme 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 diese konfiguriert sind und der Absender der Anfrage über die entsprechenden `s3:Get*` Berechtigungen verfügt.

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

Wie die Aufbewahrungseinstellungen für ein Objekt aktualisiert werden

Wenn Sie die Einstellungen für die rechtliche Aufbewahrung oder die Aufbewahrungsfristen einer bestehenden Objektversion aktualisieren müssen, können Sie die folgenden Objekt-Subressourcenoperationen durchführen:

- `PutObjectLegalHold`

  Wenn der neue Wert für den Legal Hold ON ist, wird das Objekt unter Legal Hold gesetzt. Wenn der Wert

für den Legal Hold OFF ist, wird der Legal Hold aufgehoben.

- `PutObjectRetention`
  - Der Moduswert kann COMPLIANCE oder GOVERNANCE sein (Groß-/Kleinschreibung).
  - Der Wert für die Aufbewahrungsfrist muss im Format `2020-08-10T21:46:00Z` angegeben werden. Sekundenbruchteile sind zulässig, jedoch werden nur 3 Dezimalstellen beibehalten (Millisekundenpräzision). Andere ISO 8601 Formate sind nicht zulässig.
  - Hat eine Objektversion bereits ein Gültigkeitsdatum (Retain-Until-Datum), kann dieses nur erhöht werden. Der neue Wert muss in der Zukunft liegen.

Verwendung des GOVERNANCE Modus

Benutzer, die über die `s3:BypassGovernanceRetention` Berechtigung verfügen, können die aktiven Aufbewahrungseinstellungen eines Objekts, das den GOVERNANCE-Modus verwendet, umgehen. Jede DELETE- oder PutObjectRetention Operation muss den `x-amz-bypass-governance-retention:true` Anfrageheader enthalten. Diese Benutzer können folgende zusätzliche Operationen ausführen:

- Führen Sie DeleteObject oder DeleteObjects Operationen durch, um eine Objektversion zu löschen, bevor deren Aufbewahrungsfrist abgelaufen ist.

  Objekte, die einem Legal Hold unterliegen, können nicht gelöscht werden. Legal Hold muss deaktiviert sein.

- Durchführen von PutObjectRetention-Operationen, die den Modus einer Objektversion von GOVERNANCE in COMPLIANCE ändern, bevor der Aufbewahrungszeitraum des Objekts abgelaufen ist.

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

- Sie können PutObjectRetention-Operationen durchführen, um den Aufbewahrungszeitraum einer Objektversion zu verlängern, zu verkürzen oder zu entfernen.

Verwandte Informationen

- "Objekte mit S3 Object Lock verwalten"
- "S3 Object Lock zur Aufbewahrung von Objekten verwenden"
- "Amazon Simple Storage Service Benutzerhandbuch: Objekte sperren"

Erfahren Sie mehr über die S3-Lebenszykluskonfiguration für StorageGRID

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

Das einfache Beispiel in diesem Abschnitt veranschaulicht, wie eine S3-Lifecycle-Konfiguration steuern kann, wann bestimmte Objekte aus bestimmten S3-Buckets gelöscht (abgelaufen) werden. Das Beispiel in diesem Abschnitt dient nur zur Veranschaulichung. Vollständige Informationen zur Erstellung von S3-Lifecycle-Konfigurationen finden sich "Amazon Simple Storage Service Benutzerhandbuch: Objektlebenszyklusverwaltung". Es ist zu beachten, dass StorageGRID nur Expiration-Aktionen unterstützt und keine Transition-Aktionen.

Was eine Lebenszykluskonfiguration ist

Eine Lebenszykluskonfiguration ist ein Satz von Regeln, die auf die Objekte in bestimmten S3-Buckets

angewendet werden. Jede Regel legt fest, 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: Ein Objekt wird gelöscht, wenn ein bestimmtes Datum erreicht ist oder eine bestimmte Anzahl von Tagen ab dem Zeitpunkt der Erfassung des Objekts vergangen ist.
- NoncurrentVersionExpiration: Ein Objekt wird gelöscht, wenn eine bestimmte Anzahl von Tagen erreicht ist, beginnend ab dem Zeitpunkt, an dem das Objekt nicht mehr aktuell ist.
- Filter (Präfix, Tag)
- Status
- ID

Jedes Objekt unterliegt den Aufbewahrungseinstellungen entweder eines S3-Bucket-Lebenszyklus oder einer ILM-Richtlinie. Wenn ein S3-Bucket-Lebenszyklus konfiguriert ist, überschreiben die Ablaufaktionen des Lebenszyklus die ILM-Richtlinie für Objekte, die dem Bucket-Lebenszyklusfilter entsprechen. Objekte, die dem Bucket-Lebenszyklusfilter nicht 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 gilt, dass Objektversionen unbegrenzt aufbewahrt werden. Siehe "Beispielprioritäten für den S3 Bucket-Lebenszyklus und die ILM-Richtlinie".

Daher kann ein Objekt aus dem Grid entfernt werden, obwohl die Platzierungsanweisungen in einer ILM-Regel weiterhin für das Objekt gelten. Oder ein Objekt kann im Grid verbleiben, selbst nachdem alle ILM-Platzierungsanweisungen für das Objekt abgelaufen sind. Weitere Informationen finden sich unter "Wie ILM während der gesamten Lebensdauer eines Objekts funktioniert".

> Die Bucket-Lebenszykluskonfiguration kann für Buckets verwendet werden, bei denen S3 Object Lock aktiviert ist, aber für ältere Compliant Buckets wird die Bucket-Lebenszykluskonfiguration 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 wird eine JSON-Datei erstellt, die eine oder mehrere Regeln enthält. Diese JSON-Datei enthält beispielsweise drei Regeln:

1. Regel 1 gilt nur für Objekte, die mit dem Präfix `category1/` übereinstimmen und die einen `key2` Wert von `tag2` haben. Der `Expiration` 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 `category2/` entsprechen. Der `Expiration` Parameter legt fest, 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 Objektaufnahme. Wenn das aktuelle Datum den Aufnahmezeitpunkt plus die Anzahl der Tage überschreitet, können einige Objekte aus dem Bucket entfernt werden, sobald die Lifecycle-Konfiguration angewendet wird.

3. Regel 3 gilt nur für Objekte, die dem Präfix `category3/` entsprechen. Der `Expiration` Parameter gibt an, dass alle nicht aktuellen Versionen entsprechender Objekte 50 Tage nach dem Übergang in den nicht aktuellen Status 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 Lifecycle-Konfigurationsdatei erstellt haben, wenden Sie diese auf einen Bucket an, indem Sie eine PutBucketLifecycleConfiguration-Anforderung senden.

Diese Anfrage wendet die Lebenszykluskonfiguration aus der Beispieldatei auf Objekte in einem Bucket mit dem Namen `testbucket` an.

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

Um zu validieren, dass eine Lebenszykluskonfiguration erfolgreich auf den Bucket angewendet wurde, kann eine GetBucketLifecycleConfiguration-Anfrage gestellt werden. Beispielsweise:

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

Eine erfolgreiche Antwort listet die soeben angewendete Lebenszykluskonfiguration auf.

Überprüfen, ob der Bucket-Lifecycle-Ablauf auf das Objekt angewendet wird

Es lässt sich feststellen, ob eine Ablaufregel in der Lebenszykluskonfiguration auf ein bestimmtes Objekt zutrifft, wenn eine PutObject-, HeadObject- oder GetObject-Anfrage gestellt wird. Wenn eine Regel zutrifft, enthält die Antwort einen `Expiration` Parameter, der angibt, wann das Objekt abläuft und welche Ablaufregel angewendet wurde.

> Da der Bucket-Lebenszyklus ILM überschreibt, ist das `expiry-date` angezeigte Datum das tatsächliche Löschdatum des Objekts. Weitere Informationen finden Sie unter "Wie die Objektretention bestimmt wird".

Beispielsweise wurde diese PutObject-Anfrage am 22. Juni 2020 gestellt und platziert ein Objekt im `testbucket` Bucket.

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

Die Erfolgsmeldung zeigt an, dass das Objekt in 100 Tagen (01. Okt 2020) abläuft und dass es mit Regel 2 der Lebenszykluskonfiguration übereinstimmt.

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

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

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

Die Erfolgsmeldung enthält die Metadaten des Objekts und gibt an, dass das Objekt in 100 Tagen abläuft und dass es 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": {}
}
```

Bei Buckets mit aktivierter Versionsverwaltung gilt der `x-amz-expiration` Antwortheader nur für die aktuellen Versionen der Objekte.

Empfehlungen für die Implementierung der S3 REST API mit StorageGRID

Diese Empfehlungen sind bei der Implementierung der S3 REST API zur Verwendung mit StorageGRID zu beachten.

Empfehlungen für HEADs zu nicht existierenden Objekten

Wenn Ihre Anwendung regelmäßig prüft, ob ein Objekt unter einem Pfad existiert, unter dem Sie nicht erwarten, dass das Objekt tatsächlich existiert, sollte die Konsistenz „Verfügbar“ "Konsistenz" verwendet werden. Zum Beispiel ist die Konsistenz „Verfügbar“ zu verwenden, wenn Ihre Anwendung einen HEAD-Befehl an einen Speicherort sendet, bevor per PUT darauf geschrieben wird.

Andernfalls kann es zu einer hohen Anzahl von 500 Internal Server-Fehlern kommen, wenn die HEAD-Operation das Objekt nicht findet und zwei oder mehr Storage Nodes am selben Standort nicht verfügbar sind oder ein entfernter Standort nicht erreichbar ist.

Sie können die Konsistenz „Verfügbar“ für jeden Bucket mit der "PUT Bucket Konsistenz" Anfrage festlegen oder die Konsistenz im Anfrageheader für eine einzelne API-Operation angeben.

Empfehlungen für Objektschlüssel

Diese Empfehlungen für Objektschlüsselnamen gelten abhängig vom 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 von Objektschlüsseln. Dies steht im Gegensatz zur früheren AWS-Empfehlung für Schlüsselpräfixe. Stattdessen sollten nicht zufällige, nicht eindeutige Präfixe wie `image` verwendet werden.
- Wenn Sie der früheren AWS-Empfehlung folgen, zufällige und eindeutige Zeichen in den Schlüsselpräfixen zu verwenden, sollte den Objektschlüsseln ein Verzeichnisname vorangestellt werden. Das bedeutet, folgendes Format wird verwendet:

  ```
  mybucket/mydir/f8e3-image3132.jpg
  ```

  Statt dieses Formats:

  ```
  mybucket/f8e3-image3132.jpg
  ```

Buckets, die in StorageGRID 11.4 oder höher erstellt wurden

Die Beschränkung von Objektschlüsselnamen zur Einhaltung von Performance-Best Practices ist nicht erforderlich. In den meisten Fällen können für die ersten vier Zeichen von Objektschlüsselnamen zufällige Werte verwendet werden.

> Eine Ausnahme bildet ein S3-Workload, der kontinuierlich alle Objekte nach kurzer Zeit entfernt. Um die Leistungseinbußen für diesen Anwendungsfall zu minimieren, sollte ein führender Teil des Schlüsselnamens alle paar tausend Objekte mit etwas wie dem Datum variiert werden. Beispielsweise schreibt ein S3-Client typischerweise 2.000 Objekte pro Sekunde und die ILM- oder Bucket-Lifecycle-Richtlinie entfernt alle Objekte nach drei Tagen. Um die Leistungseinbußen zu minimieren, könnte die Benennung der Schlüssel nach folgendem Muster erfolgen: `/mybucket/mydir/yyyymmddhhmmss-random_UUID.jpg`

Empfehlungen für „Range Reads“

Wenn die "Globale Option zum Komprimieren gespeicherter Objekte" Option aktiviert ist, sollten S3-Clientanwendungen GetObject-Operationen vermeiden, die einen Bytebereich zur Rückgabe festlegen. Diese „Bereichslese“-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 50 GB großen komprimierten Objekt zu lesen.

Wenn Bereiche aus komprimierten Objekten gelesen werden, kann die Clientanfrage eine Zeitüberschreitung aufweisen.

> Wenn Objekte komprimiert werden müssen und die Clientanwendung Bereichslesevorgänge verwenden muss, sollte das Zeitüberschreitung für die Anwendung erhöht werden.

Unterstützung für die Amazon S3 REST API

Unterstützte Operationen und Einschränkungen der S3 REST API in StorageGRID

Das StorageGRID System implementiert die Simple Storage Service API (API Version 2006-03-01) mit Unterstützung für die meisten Operationen, jedoch mit einigen Einschränkungen. Beim Integrieren von S3 REST API Clientanwendungen sind die Implementierungsdetails zu berücksichtigen.

Das StorageGRID System unterstützt sowohl virtuelle Hosted-Style-Anfragen als auch Path-Style-Anfragen.

Datumsverarbeitung

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

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

Gängige Anfrage-Header

Das StorageGRID System unterstützt die allgemeinen Anforderungsheader, die von "Amazon Simple Storage Service API Reference: Allgemeine Anfrageheader" definiert wurden, mit einer Ausnahme.

Anfrageheader	Implementierung
Autorisierung	Volle Unterstützung für AWS Signature Version 2 Unterstützung für AWS Signature Version 4, mit folgenden Ausnahmen: • Wenn Sie den tatsächlichen Nutzlast Prüfsumme-Wert in `x-amz-content-sha256` angeben, wird der Wert ohne Validierung akzeptiert, so als ob der Wert `UNSIGNED-PAYLOAD` für den Header angegeben worden wäre. Wenn ein `x-amz-content-sha256` Header-Wert angegeben wird, der `aws-chunked` Streaming impliziert (zum Beispiel STREAMING-AWS4-HMAC-SHA256-PAYLOAD), werden die Chunk-Signaturen nicht mit den Chunk-Daten überprüft.

Allgemeine Antwort-Header

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

Antwort-Header	Implementierung
x-amz-id-2	Nicht verwendet

Wie StorageGRID S3 REST API Anforderungen authentifiziert

Das StorageGRID System unterstützt sowohl authentifizierten als auch anonymen Zugriff auf Objekte über die S3 API.

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

Authentifizierte Anfragen müssen mit Ihrer Access Key ID und Ihrem Secret Access Key signiert werden.

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

Der HTTP Authorization-Header wird verwendet.

Der HTTP `Authorization` Header wird von allen S3 API-Operationen verwendet, außer bei anonymen Anfragen, sofern diese durch die Bucket-Richtlinie zugelassen sind. Der `Authorization` Header enthält alle erforderlichen Signaturinformationen zur Authentifizierung einer Anfrage.

Abfrageparameter verwenden

Sie können Abfrageparameter verwenden, um einer URL Authentifizierungsinformationen hinzuzufügen. Dies wird als Vorsignieren der URL bezeichnet, was einen temporären Zugriff auf bestimmte Ressourcen ermöglicht. Benutzer mit der vorsignierten URL müssen den geheimen Zugriffsschlüssel nicht kennen, um auf die Ressource zuzugreifen, wodurch ein eingeschränkter Zugriff für Dritte auf eine Ressource ermöglicht wird.

Unterstützte Serviceoperationen in StorageGRID

Das StorageGRID System unterstützt die folgenden Operationen für den Dienst.

Betrieb	Implementierung
ListBuckets (früher als GET Service bezeichnet)	Implementiert mit dem gesamten Verhalten der Amazon S3 REST API. Änderungen vorbehalten.
Speichernutzung abrufen	Die StorageGRID "Speichernutzung abrufen" Anfrage gibt die gesamte vom Konto und von jedem mit dem Konto verknüpften Bucket genutzte Speichermenge an. Dies ist eine Operation für den Service mit dem Pfad / und einem benutzerdefinierten Abfrageparameter (`?x-ntap-sg-usage` hinzugefügt.
OPTIONEN /	Clientanwendungen können `OPTIONS /`Anfragen an den S3-Port eines Storage Node senden, ohne S3-Authentifizierungsdaten bereitzustellen, um festzustellen, ob der Storage Node verfügbar ist. Diese Anfrage kann für die Überwachung verwendet werden oder um externen Load Balancern zu ermöglichen, zu erkennen, wenn ein Storage Node ausgefallen ist.

S3-Bucket-Operationen und Implementierungsdetails in StorageGRID

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

Jedes Grid kann maximal 100.000 Buckets enthalten.

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

Die Beschränkungen für Bucket-Namen orientieren sich an den Beschränkungen der AWS-Region US Standard, sollten aber zusätzlich auf DNS-Namenskonventionen beschränkt werden, um S3 Virtual Hosted-Style-Anfragen zu unterstützen.

Weitere Informationen sind im Folgenden verfügbar:

- "Amazon Simple Storage Service Benutzerhandbuch: Bucket-Kontingente, Einschränkungen und Begrenzungen"
- "S3-Endpunkt-Domänennamen konfigurieren"

Die ListObjects (GET Bucket) und ListObjectVersions (GET Bucket object versions) Operationen unterstützen StorageGRID "Konsistenzwerte".

Sie können überprüfen, ob Aktualisierungen der letzten Zugriffszeit für einzelne Buckets aktiviert oder deaktiviert sind. Siehe "Letzte Zugriffszeit des Buckets abrufen".

Die folgende Tabelle beschreibt, wie StorageGRID S3 REST API-Bucket-Operationen implementiert. Um eine dieser Operationen durchzuführen, müssen die erforderlichen Zugangsdaten für das Konto bereitgestellt werden.

Betrieb	Implementierung
CreateBucket	Erstellt einen neuen Bucket. Durch das Erstellen des Buckets werden Sie dessen Besitzer. • Bucket-Namen müssen die folgenden Regeln erfüllen: ◦ Muss in jedem StorageGRID System eindeutig sein (nicht nur innerhalb des Mandantenkontos). ◦ Muss DNS-konform sein. ◦ Muss mindestens 3 und höchstens 63 Zeichen enthalten. ◦ Es kann sich um eine Reihe von einem oder mehreren Labels handeln, wobei benachbarte Labels durch einen Punkt getrennt sind. Jedes Label 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 virtuellen Hosting-Stil sollten keine Punkte verwendet werden. Punkte verursachen Probleme bei der Überprüfung des Server-Wildcard-Zertifikats. • Standardmäßig werden Buckets in der `us-east-1` Region erstellt; es kann jedoch das `LocationConstraint` Anfrageelement im Anfragetext verwendet werden, um eine andere Region anzugeben. Bei Verwendung des `LocationConstraint` Elements muss der genaue Name einer Region angegeben werden, die mit dem Grid Manager oder der Grid Management API definiert wurde. Falls der Name der zu verwendenden Region nicht bekannt ist, empfiehlt sich die Kontaktaufnahme mit dem Systemadministrator. **Hinweis**: Es tritt ein Fehler auf, wenn Ihre CreateBucket-Anfrage eine Region verwendet, die in StorageGRID nicht definiert wurde. • Sie können den `x-amz-bucket-object-lock-enabled` Anforderungsheader einfügen, um einen Bucket mit aktivierter S3 Object Lock zu erstellen. Siehe "S3 REST API zur Konfiguration von S3 Object Lock verwenden". Sie müssen S3 Object Lock beim Erstellen des Buckets aktivieren. Sie können S3 Object Lock nach dem Erstellen eines Buckets weder hinzufügen noch deaktivieren. S3 Object Lock erfordert die Bucket-Versionierung, die beim Erstellen des Buckets automatisch aktiviert wird.
DeleteBucket	Löscht den Bucket.
DeleteBucketCors	Löscht die CORS Konfiguration für den Bucket.
DeleteBucketEncryption	Löscht die Standardverschlüsselung des Buckets. Bereits verschlüsselte Objekte bleiben verschlüsselt, aber neu hinzugefügte Objekte werden nicht verschlüsselt.

Betrieb	Implementierung
DeleteBucketLifecycle	Löscht die Lebenszykluskonfiguration aus dem Bucket. Siehe "S3 Lifecycle-Konfiguration erstellen".
DeleteBucketPolicy	Löscht die dem Bucket zugeordnete Richtlinie.
DeleteBucketReplication	Löscht die dem Bucket zugeordnete Replikationskonfiguration.
DeleteBucketTagging	Verwendet die `tagging` Subressource, um alle Tags aus einem Bucket zu entfernen. **Achtung**: Wenn für diesen Bucket ein nicht standardmäßiges ILM-Policy-Tag festgelegt ist, gibt es ein `NTAP-SG-ILM-BUCKET-TAG` Bucket-Tag mit einem zugewiesenen Wert. Es sollte keine DeleteBucketTagging-Anfrage gestellt werden, wenn ein `NTAP-SG-ILM-BUCKET-TAG` Bucket-Tag vorhanden ist. Stattdessen sollte eine PutBucketTagging-Anfrage mit nur dem `NTAP-SG-ILM-BUCKET-TAG` Tag und dessen zugewiesenem Wert gestellt werden, um alle anderen Tags aus dem Bucket zu entfernen. Das `NTAP-SG-ILM-BUCKET-TAG` Bucket-Tag sollte nicht geändert oder entfernt werden.
GetBucketAcl	Gibt eine positive Antwort sowie die ID, DisplayName und die Berechtigung des Bucket-Inhabers zurück, was darauf hinweist, dass der Inhaber vollen Zugriff auf den Bucket hat.
GetBucketCors	Gibt die `cors`Konfiguration für den Bucket zurück.
GetBucketEncryption	Gibt die Standardverschlüsselungskonfiguration für den Bucket zurück.
GetBucketLifecycleConfiguration (früher als GET Bucket Lifecycle bezeichnet)	Gibt die Lebenszykluskonfiguration für den Bucket zurück. Siehe "S3 Lifecycle-Konfiguration erstellen".
GetBucketLocation	Gibt die Region zurück, die mithilfe des `LocationConstraint` Elements in der CreateBucket-Anfrage festgelegt wurde. Wenn die Region des Buckets `us-east-1` ist, wird eine leere Zeichenkette für die Region zurückgegeben.
GetBucketNotificationConfiguration (früher als GET Bucket notification bezeichnet)	Gibt die dem Bucket zugeordnete Benachrichtigungskonfiguration zurück.
GetBucketPolicy	Gibt die dem Bucket zugeordnete Richtlinie zurück.
GetBucketReplication	Gibt die dem Bucket zugeordnete Replikationskonfiguration zurück.

Betrieb	Implementierung
GetBucketTagging	Verwendet die `tagging` Subressource, um alle Tags für einen Bucket zurückzugeben. **Achtung**: Wenn für diesen Bucket ein nicht standardmäßiges ILM-Policy-Tag festgelegt ist, gibt es ein `NTAP-SG-ILM-BUCKET-TAG` Bucket-Tag mit einem zugewiesenen Wert. Dieses Tag darf nicht geändert oder entfernt werden.
GetBucketVersioning	Diese Implementierung verwendet die `versioning` Subressource, um den Versionsstatus eines Buckets zurückzugeben. • *blank*: Die Versionierung wurde nie aktiviert (Bucket ist "Unversioned") • Aktiviert: Versionsverwaltung ist aktiviert • Ausgesetzt: Die Versionierung war zuvor aktiviert und ist ausgesetzt
GetObjectLockConfiguration	Gibt den standardmäßigen Aufbewahrungsmodus und die standardmäßige Aufbewahrungsdauer des Buckets zurück, sofern konfiguriert. Siehe "S3 REST API zur Konfiguration von S3 Object Lock verwenden".
HeadBucket	Prüft, ob ein Bucket existiert und ob Sie die Berechtigung haben, darauf zuzugreifen. Diese Operation 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.

Betrieb	Implementierung
ListObjects und ListObjectsV2 (früher als GET Bucket bezeichnet)	Gibt einige oder alle (bis zu 1.000) Objekte in einem Bucket zurück. Die Speicherklasse für Objekte kann einen von zwei Werten annehmen, selbst wenn das Objekt mit der `REDUCED_REDUNDANCY` Speicherklasse-Option importiert wurde: • `STANDARD`, was darauf hinweist, dass das Objekt in einem Speicherpool aus Storage Nodes gespeichert ist. • `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 auch einige `CommonPrefixes` enthalten, die keine Schlüssel enthalten. Für die HeadObject- und ListObject-Anfragen gibt StorageGRID die LastModified-Zeitstempel mit unterschiedlicher Genauigkeit zurück, während AWS die Zeitstempel mit derselben Genauigkeit zurückgibt, wie in den folgenden Beispielen gezeigt: • StorageGRID HeadObject: "LastModified": "2024-09-26T16:43:24+00:00" • StorageGRID ListObject: "LastModified": "2024-09-26T16:43:24.931000+00:00" • AWS HeadObject: "LastModified": "2023-10-17T00:19:54+00:00" • AWS ListObject: "LastModified": "2023-10-17T00:19:54+00:00"
ListObjectVersions (früher als GET Bucket Object versions bezeichnet)	Bei Lesezugriff auf einen Bucket listet diese Operation mit der `versions` Unterressource die Metadaten aller Versionen von Objekten im Bucket auf.
PutBucketCors	Setzt die CORS-Konfiguration für einen Bucket, sodass der Bucket ursprungsübergreifende Anfragen bedienen kann. Cross-Origin Resource Sharing (CORS) ist ein Sicherheitsmechanismus, der es Client-Webanwendungen in einer Domäne ermöglicht, auf Ressourcen in einer anderen Domäne zuzugreifen. Beispielsweise wird ein S3-Bucket namens `images` verwendet, um Grafiken zu speichern. Durch das Setzen der CORS-Konfiguration für den `images` Bucket kann ermöglicht werden, dass die Bilder in diesem Bucket auf der Website `http://www.example.com` angezeigt werden.

Betrieb	Implementierung
PutBucketEncryption	Legt den Standardverschlüsselungsstatus eines vorhandenen Buckets fest. Wenn die Bucket-Verschlüsselung aktiviert ist, werden alle neu hinzugefügten Objekte verschlüsselt. StorageGRID unterstützt serverseitige Verschlüsselung mit von StorageGRID verwalteten Schlüsseln. Beim Festlegen der Konfigurationsregel für die serverseitige Verschlüsselung ist der `SSEAlgorithm` Parameter auf `AES256` zu setzen, und der `KMSMasterKeyID` Parameter darf nicht verwendet werden. Die Standardverschlüsselungskonfiguration des Buckets wird ignoriert, wenn die Objekt-Upload-Anforderung bereits eine Verschlüsselung vorsieht (d. h. wenn die Anforderung den `x-amz-server-side-encryption-*` request header enthält).
PutBucketLifecycleConfiguration (früher als PUT Bucket lifecycle bezeichnet)	Erstellt eine neue Lebenszykluskonfiguration für den Bucket oder ersetzt eine bestehende 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 • ID StorageGRID unterstützt diese Aktionen nicht: • AbortIncompleteMultipartUpload • Übergang Siehe "S3 Lifecycle-Konfiguration erstellen". Weitere Informationen darüber, wie die Ablaufaktion in einem Bucket-Lebenszyklus mit ILM-Platzierungsanweisungen zusammenwirkt, finden Sie unter "Wie ILM während der gesamten Lebensdauer eines Objekts funktioniert". **Hinweis**: Die Bucket-Lebenszykluskonfiguration kann mit Buckets verwendet werden, bei denen S3 Object Lock aktiviert ist, aber für ältere Compliant Buckets wird die Bucket-Lebenszykluskonfiguration nicht unterstützt.

Betrieb	Implementierung

PutBucketNotificationConfiguration
 	
(früher als PUT Bucket notification bezeichnet)
	Konfiguriert Benachrichtigungen für den Bucket mithilfe der im Anfragetext enthaltenen Benachrichtigungskonfigurations-XML. Es sind die folgenden Implementierungsdetails zu berücksichtigen:
	
	• StorageGRID unterstützt Amazon Simple Notification Service (Amazon SNS) Topics, Kafka Topics oder Webhook-Endpunkte als Ziele. Simple Queue Service (SQS) oder AWS Lambda Endpunkte werden nicht unterstützt.
	• Als Ziel für Benachrichtigungen muss die 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 mit dem Code `InvalidArgument` zurückgegeben.
	
	• Für die folgenden Ereignistypen können keine Benachrichtigungen konfiguriert werden. Diese Ereignistypen werden **nicht** unterstützt.
	◦ `s3:ReducedRedundancyLostObject`
	◦ `s3:ObjectRestore:Completed`
	• Von StorageGRID gesendete Ereignisbenachrichtigungen verwenden das Standard-JSON-Format, enthalten jedoch einige Schlüssel nicht und verwenden für andere spezifische Werte, wie in der folgenden Liste dargestellt:
	◦ **eventSource**
	
	`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. Siehe "Bucket- und Gruppenzugriffsrichtlinien".

Betrieb	Implementierung
PutBucketReplication	Konfiguriert "StorageGRID CloudMirror Replikation" für den Bucket mithilfe der im Anfragetext bereitgestellten Replikationskonfigurations-XML. Für die CloudMirror-Replikation sind die folgenden Implementierungsdetails zu beachten: • StorageGRID unterstützt ausschließlich die Replikationskonfiguration V1. Das bedeutet, dass StorageGRID die Verwendung des `Filter` Elements für Regeln nicht unterstützt und die V1-Konventionen für das Löschen von Objektversionen anwendet. Weitere Informationen finden sich 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 Replikationskonfigurations-XML einen anderen Ziel-Bucket angeben. Ein Quell-Bucket kann in mehr als einen Ziel-Bucket replizieren. • Ziel-Buckets müssen als URN von StorageGRID Endpunkten angegeben werden, wie im Tenant Manager oder der Tenant Management API definiert. Siehe "Konfiguration der CloudMirror Replikation". Für eine erfolgreiche Replikationskonfiguration muss der Endpunkt vorhanden sein. Existiert der Endpunkt nicht, schlägt die Anfrage als `400 Bad Request` fehl. Die Fehlermeldung lautet: `Unable to save the replication policy. The specified endpoint URN does not exist: URN.` • Sie müssen in der Konfigurations-XML keinen `Role` angeben. Dieser Wert wird von StorageGRID nicht verwendet und bei der Übermittlung ignoriert. • Wenn die Speicherklasse in der Konfigurations-XML weggelassen wird, verwendet StorageGRID standardmäßig die `STANDARD` Speicherklasse. • Wenn ein Objekt aus dem Quell-Bucket gelöscht wird oder der Quell-Bucket selbst gelöscht wird, verhält sich die regionsübergreifende Replikation wie folgt: ◦ Wenn das Objekt oder der Bucket gelöscht wird, bevor es repliziert wurde, wird das Objekt/der Bucket nicht repliziert und eine Benachrichtigung erfolgt nicht. ◦ Wenn das Objekt oder der Bucket nach der Replikation gelöscht wird, folgt StorageGRID dem Standard-Löschverhalten von Amazon S3 für V1 der regionsübergreifenden Replikation.

Betrieb	Implementierung
PutBucketTagging	Verwendet die `tagging` Unterressource, um eine Gruppe von Tags für einen Bucket hinzuzufügen oder zu aktualisieren. Beim Hinzufügen von Bucket-Tags sind die folgenden Einschränkungen zu beachten: • Sowohl StorageGRID als auch Amazon S3 unterstützen bis zu 50 Tags pro Bucket. • Tags, die einem Bucket zugeordnet sind, müssen eindeutige Tag-Schlüssel besitzen. Ein Tag-Schlüssel darf maximal 128 Unicode-Zeichen lang sein. • Tag-Werte können bis zu 256 Unicode-Zeichen lang sein. • Schlüssel und Werte sind Groß-/Kleinschreibung. **Achtung**: Wenn für diesen Bucket ein nicht standardmäßiges ILM-Policy-Tag festgelegt ist, gibt es ein `NTAP-SG-ILM-BUCKET-TAG` Bucket-Tag mit einem zugewiesenen Wert. Es ist sicherzustellen, dass das `NTAP-SG-ILM-BUCKET-TAG` Bucket-Tag mit dem zugewiesenen Wert in allen PutBucketTagging-Anfragen enthalten ist. Dieses Tag darf nicht geändert oder entfernt werden. **Hinweis**: Bei diesem Vorgang werden alle bereits vorhandenen Tags des Buckets überschrieben. Wenn vorhandene Tags im Satz fehlen, werden diese Tags aus dem Bucket entfernt.
PutBucketVersioning	Verwendet die `versioning` Subressource, um den Versionsstatus eines vorhandenen Buckets festzulegen. Der Versionsstatus kann mit einem der folgenden Werte festgelegt werden: • Aktiviert: Aktiviert die Versionierung für die Objekte im Bucket. Alle dem Bucket hinzugefügten Objekte erhalten eine eindeutige Versions-ID. • Ausgesetzt: Deaktiviert die Versionierung der Objekte im Bucket. Alle dem Bucket hinzugefügten Objekte erhalten die Versions-ID `null`.
PutObjectLockConfiguration	Konfiguriert oder entfernt den Standardaufbewahrungsmodus und die Standardaufbewahrungsdauer des Buckets. Wird die Standardaufbewahrungsfrist geändert, bleibt das Aufbewahrungsdatum für bestehende Objektversionen unverändert und wird nicht anhand der neuen Standardaufbewahrungsfrist neu berechnet. Siehe "S3 REST API zur Konfiguration von S3 Object Lock verwenden" für detaillierte Informationen.

Operationen an Objekten

Wie StorageGRID S3 REST API-Operationen für Objekte implementiert

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

Für alle Objektoperationen gelten folgende Bedingungen:

- StorageGRID "Konsistenzwerte" wird von allen Operationen auf Objekten unterstützt, mit Ausnahme der folgenden:
  - GetObjectAcl
  - `OPTIONS /`
  - PutObjectLegalHold
  - PutObjectRetention
  - SelectObjectContent
- Konfliktierende Clientanfragen, beispielsweise wenn zwei Clients auf denselben Schlüssel schreiben, werden nach dem Prinzip „latest-wins“ aufgelöst. Der Zeitpunkt der „latest-wins“-Auswertung richtet sich danach, wann das StorageGRID System die jeweilige Anfrage abgeschlossen hat, und nicht danach, wann S3 Clients eine Operation starten.
- Alle Objekte in einem StorageGRID Bucket gehören dem Bucket-Inhaber, einschließlich Objekten, die von einem anonymen Benutzer oder einem anderen Konto erstellt wurden.

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

Betrieb	Implementierung
DeleteObject	Bei der Verarbeitung einer DeleteObject-Anfrage versucht StorageGRID, alle Kopien des Objekts umgehend von allen Speicherorten zu entfernen. Wenn dies erfolgreich ist, gibt StorageGRID sofort eine Antwort an den Client zurück. Können nicht alle Kopien innerhalb von 30 Sekunden entfernt werden (zum Beispiel weil ein Speicherort vorübergehend nicht verfügbar ist), werden die Kopien von StorageGRID zur Entfernung in die Warteschlange gestellt und dem Client wird anschließend der Erfolg gemeldet. **Einschränkungen** • Multi-Faktor-Authentifizierung (MFA) und der Antwortheader `x-amz-mfa` werden nicht unterstützt. • Die `If-None` und `If-None-Match` Header werden akzeptiert, sind aber nicht funktionsfähig. **Versionierung** Um eine bestimmte Version zu entfernen, muss der Anfragende der Bucket-Inhaber sein und die `versionId` Unterressource verwenden. Die Verwendung dieser Unterressource löscht die Version dauerhaft. Wenn die `versionId` Unterressource einem Löschmarker entspricht, wird der Antwortheader `x-amz-delete-marker` mit dem Wert `true` zurückgegeben. • Wenn ein Objekt ohne die `versionId` Unterressource in einem Bucket mit aktivierter Versionierung gelöscht wird, wird eine Löschmarkierung generiert. Die `versionId` für die Löschmarkierung wird über den `x-amz-version-id` Antwortheader zurückgegeben, und der `x-amz-delete-marker` Antwortheader wird mit `true` zurückgegeben. • Wird ein Objekt ohne die `versionId` Unterressource in einem Bucket mit deaktivierter Versionierung gelöscht, erfolgt eine endgültige Entfernung einer bereits vorhandenen „null“-Version oder eines „null“-Löschmarkers sowie die Erzeugung eines neuen „null“-Löschmarkers. Der `x-amz-delete-marker` Response-Header wird mit dem Wert `true` zurückgegeben. **Hinweis**: In bestimmten Fällen können für ein Objekt mehrere Löschmarkierungen vorhanden sein. Siehe "S3 REST API zur Konfiguration von S3 Object Lock verwenden", um zu erfahren, wie Objektversionen im GOVERNANCE-Modus gelöscht werden.

Betrieb	Implementierung
DeleteObjects (früher als DELETE Multiple Objects bezeichnet)	Multi-Faktor-Authentifizierung (MFA) und der Antwortheader `x-amz-mfa` werden nicht unterstützt. In derselben Anforderungsnachricht können mehrere Objekte gelöscht werden. Siehe "S3 REST API zur Konfiguration von S3 Object Lock verwenden", um zu erfahren, wie Objektversionen im GOVERNANCE-Modus gelöscht werden.
DeleteObjectTagging	Verwendet die `tagging` Subressource, um alle Tags von einem Objekt zu entfernen. **Versionierung** Wenn der `versionId` Abfrageparameter in der Anfrage nicht angegeben ist, werden durch den Vorgang alle Tags der neuesten Version des Objekts in einem versionierten Bucket gelöscht. Wenn die aktuelle Version des Objekts eine Löschmarkierung ist, wird ein "MethodNotAllowed"-Status zurückgegeben, wobei der `x-amz-delete-marker` Antwortheader auf `true` gesetzt ist.
GetObject	"GetObject"
GetObjectAcl	Wenn die erforderlichen Zugangsdaten für das Konto bereitgestellt werden, gibt die Operation eine positive Antwort sowie die ID, DisplayName und die Berechtigung des Objektinhabers zurück, was darauf hinweist, dass der Inhaber vollen Zugriff auf das Objekt hat.
GetObjectLegalHold	"S3 REST API zur Konfiguration von S3 Object Lock verwenden"
GetObjectRetention	"S3 REST API zur Konfiguration von S3 Object Lock verwenden"
GetObjectTagging	Verwendet die `tagging` Subressource, um alle Tags für ein Objekt zurückzugeben. **Versionierung** Wird der `versionId` Abfrageparameter in der Anfrage nicht angegeben, gibt der Vorgang alle Tags der aktuellsten Version des Objekts in einem versionierten Bucket zurück. Wenn die aktuelle Version des Objekts eine Löschmarkierung ist, wird ein "MethodNotAllowed"-Status zurückgegeben, wobei der `x-amz-delete-marker` Antwortheader auf `true` gesetzt ist.
HeadObject	"HeadObject"
RestoreObject	"RestoreObject"

Betrieb	Implementierung
PutObject	"PutObject"
CopyObject (früher als PUT Object - Copy bezeichnet)	"CopyObject"
PutObjectLegalHold	"S3 REST API zur Konfiguration von S3 Object Lock verwenden"
PutObjectRetention	"S3 REST API zur Konfiguration von S3 Object Lock verwenden"

Betrieb	Implementierung
PutObjectTagging	Verwendet die `tagging` Subressource, um einem bestehenden Objekt eine Reihe von Tags hinzuzufügen. **Objekt-Tag-Grenzwerte** Sie können neuen Objekten beim Hochladen Tags hinzufügen oder sie bestehenden Objekten hinzufügen. Sowohl StorageGRID als auch Amazon S3 unterstützen bis zu 10 Tags pro Objekt. Tags, die einem Objekt zugeordnet sind, 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. Schlüssel und Werte sind Groß-/Kleinschreibung. **Tag-Aktualisierungen und Ingest-Verhalten** Wenn Sie PutObjectTagging verwenden, um die Tags eines Objekts zu aktualisieren, wird das Objekt von StorageGRID nicht erneut aufgenommen. Das bedeutet, dass die in der entsprechenden ILM-Regel festgelegte Option für das Aufnahmeverhalten nicht verwendet wird. Alle durch die Aktualisierung ausgelösten Änderungen an der Objektplatzierung erfolgen, wenn ILM durch die normalen Hintergrundprozesse neu ausgewertet wird. Das bedeutet, dass bei Verwendung der Option „Streng“ für das Aufnahmeverhalten in der ILM-Regel keine Aktion ausgeführt wird, wenn die erforderlichen Objektplatzierungen nicht möglich sind (zum Beispiel, weil ein neu benötigter Speicherort nicht verfügbar ist). Das aktualisierte Objekt behält seine aktuelle Platzierung bei, bis die erforderliche Platzierung möglich ist. **Konflikte beheben** Konfliktierende Clientanfragen, beispielsweise wenn zwei Clients auf denselben Schlüssel schreiben, werden nach dem Prinzip „latest-wins“ aufgelöst. Der Zeitpunkt der „latest-wins“-Auswertung richtet sich danach, wann das StorageGRID System die jeweilige Anfrage abgeschlossen hat, und nicht danach, wann S3 Clients eine Operation starten. **Versionierung** Wird der `versionId` Abfrageparameter in der Anfrage nicht angegeben, fügt der Vorgang Tags zur aktuellsten Version des Objekts in einem versionierten Bucket hinzu. Wenn die aktuelle Version des Objekts eine Löschmarkierung ist, wird ein "MethodNotAllowed"-Status zurückgegeben, wobei der `x-amz-delete-marker` Antwortheader auf `true` gesetzt ist.
SelectObjectContent	"SelectObjectContent"

Unterstützte Amazon S3 Select-Operationen in StorageGRID

StorageGRID unterstützt die folgenden Amazon S3 Select Klauseln, Datentypen und Operatoren für die "SelectObjectContent Befehl".

Alle nicht aufgeführten Artikel sind nicht unterstützt.

Informationen zur Syntax sind unter "SelectObjectContent" zu finden. Weitere Informationen zu S3 Select sind unter "AWS Dokumentation für S3 Select" zu finden.

Nur Mandantenkonten, für die S3 Select aktiviert ist, können SelectObjectContent-Abfragen ausführen. Siehe "Überlegungen und Anforderungen für die Verwendung von S3 Select".

${post_edited_translations.segment}

- ${post_edited_translations.segment}
- FROM-Klausel
- ${post_edited_translations.segment}
- LIMIT Klausel

${post_edited_translations.segment}

- bool
- ${post_edited_translations.segment}
- Zeichenkette
- ${post_edited_translations.segment}
- ${post_edited_translations.segment}
- Zeitstempel

Betreiber

Logische Operatoren

- ${post_edited_translations.segment}
- ${post_edited_translations.segment}
- ${post_edited_translations.segment}

Vergleichsoperatoren

- <
- >
- ⇐
- >=
- =
- =
- <>
- ${post_edited_translations.segment}
- ${post_edited_translations.segment}
- ${post_edited_translations.segment}

Mustervergleichsoperatoren

- WIE
- _
- %

${post_edited_translations.segment}

- ${post_edited_translations.segment}
- ${post_edited_translations.segment}

${post_edited_translations.segment}

- +
- -
- *
- /
- %

StorageGRID folgt der Operatorrangfolge von Amazon S3 Select.

Aggregatfunktionen

- ${post_edited_translations.segment}
- ${post_edited_translations.segment}
- ${post_edited_translations.segment}
- ${post_edited_translations.segment}
- SUMME()

Bedingte Funktionen

- ${post_edited_translations.segment}
- ${post_edited_translations.segment}
- ${post_edited_translations.segment}

Konvertierungsfunktionen

- CAST (für unterstützten Datentyp)

Datumsfunktionen

- ${post_edited_translations.segment}
- DATE_DIFF
- EXTRAKT
- ${post_edited_translations.segment}
- TO_TIMESTAMP

- UTCNOW

${post_edited_translations.segment}

- ${post_edited_translations.segment}
- UNTERE
- Substring
- TRIM
- OBERE

Wie StorageGRID die serverseitige Verschlüsselung verwendet

Serverseitige Verschlüsselung ermöglicht den Schutz Ihrer Objektdaten im Ruhezustand. StorageGRID verschlüsselt die Daten beim Schreiben des Objekts und entschlüsselt die Daten beim Zugriff auf das Objekt.

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

- **SSE (serverseitige Verschlüsselung mit von StorageGRID verwalteten Schlüsseln)**: Wenn eine S3-Anfrage zum Speichern eines Objekts gestellt wird, verschlüsselt StorageGRID das Objekt mit einem eindeutigen Schlüssel. Wenn eine S3-Anfrage zum Abrufen des Objekts gestellt wird, verwendet StorageGRID den gespeicherten Schlüssel, um das Objekt zu entschlüsseln.
- **SSE-C (serverseitige Verschlüsselung mit kundenseitig bereitgestellten Schlüsseln)**: Wenn eine S3-Anfrage zum Speichern eines Objekts gestellt wird, wird ein eigener Verschlüsselungsschlüssel bereitgestellt. Beim Abrufen eines Objekts wird derselbe Verschlüsselungsschlüssel als Teil der Anfrage bereitgestellt. Wenn die beiden Verschlüsselungsschlüssel übereinstimmen, wird das Objekt entschlüsselt und die Objektdaten werden zurückgegeben.

  StorageGRID übernimmt zwar alle Verschlüsselungs- und Entschlüsselungsvorgänge für Objekte, die Verwaltung der von Ihnen bereitgestellten Verschlüsselungsschlüssel obliegt Ihnen jedoch.

  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.

${post_edited_translations.segment}

Um ein Objekt mit einem von StorageGRID verwalteten eindeutigen Schlüssel zu verschlüsseln, wird der folgende Anforderungsheader verwendet:

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

Der SSE-Anforderungsheader wird von den folgenden Objektoperationen unterstützt:

- "PutObject"

- "CopyObject"
- "CreateMultipartUpload"

${post_edited_translations.segment}

Um ein Objekt mit einem eindeutigen Schlüssel, den Sie verwalten, zu verschlüsseln, werden drei Anfrage-Header verwendet:

Anfrageheader	Beschreibung
`x-amz-server-side-encryption-customer-algorithm`	Geben Sie den Verschlüsselungsalgorithmus an. Der Header-Wert muss `AES256` sein.
`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-kodiert 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 mit 128 Bit sein.

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

- "GetObject"
- "HeadObject"
- "PutObject"
- "CopyObject"
- "CreateMultipartUpload"
- "UploadPart"
- "UploadPartCopy"

Überlegungen zur Verwendung von serverseitiger Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C)

Vor der Verwendung von SSE-C sind die folgenden Punkte zu beachten:

- ${post_edited_translations.segment}

> StorageGRID lehnt alle Anfragen über HTTP ab, wenn SSE-C verwendet wird. Aus Sicherheitsgründen ist jeder versehentlich über HTTP gesendete Schlüssel als kompromittiert zu betrachten. Der Schlüssel sollte verworfen und gegebenenfalls rotiert werden.

- ${post_edited_translations.segment}
- Sie müssen die Zuordnung von Verschlüsselungsschlüsseln zu Objekten verwalten. StorageGRID speichert keine Verschlüsselungsschlüssel. Sie sind für die Verfolgung 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.
- ${post_edited_translations.segment}

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

- Wenn die Grid-übergreifende Replikation oder die CloudMirror Replikation für den Bucket konfiguriert ist, können Sie keine SSE-C-Objekte aufnehmen. Der Ingest-Vorgang schlägt fehl.

Verwandte Informationen

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

Erstellen Sie eine Kopie eines Objekts in StorageGRID mit der S3 CopyObject-Anfrage

Die S3 CopyObject-Anfrage kann verwendet werden, um eine Kopie eines bereits in S3 gespeicherten Objekts zu erstellen. Eine CopyObject-Operation entspricht der Ausführung von GetObject gefolgt von PutObject.

Konflikte beheben

Konfliktierende Clientanfragen, beispielsweise wenn zwei Clients auf denselben Schlüssel schreiben, werden nach dem Prinzip „latest-wins“ aufgelöst. Der Zeitpunkt der „latest-wins“-Auswertung richtet sich danach, wann das StorageGRID System die jeweilige Anfrage abgeschlossen hat, und nicht danach, wann S3 Clients eine Operation starten.

Objektgröße

Die maximal empfohlene Größe für einen einzelnen PutObject-Vorgang beträgt 5 GiB (5.368.709.120 Byte). Bei Objekten, die größer als 5 GiB sind, sollte stattdessen "mehrteiliger Upload" verwendet werden.

Die maximal *unterstützte* Größe für eine einzelne PutObject-Operation beträgt 5 TiB (5.497.558.138.880 Bytes).

> Wenn Sie von StorageGRID 11.6 oder einer älteren Version aktualisiert haben, wird die Warnung „S3 PUT Object size too large“ ausgelöst, wenn Sie versuchen, ein Objekt mit mehr als 5 GiB hochzuladen. Bei einer Neuinstallation von StorageGRID 11.7 oder 11.8 wird die Warnung in diesem Fall nicht ausgelöst. Um jedoch dem AWS S3-Standard zu entsprechen, werden zukünftige Versionen von StorageGRID keine Uploads von Objekten mit mehr als 5 GiB unterstützen.

UTF-8-Zeichen in Benutzermetadaten

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

StorageGRID analysiert oder interpretiert keine maskierten UTF-8-Zeichen, die im Schlüsselname oder -wert benutzerdefinierter Metadaten enthalten sind. Maskierte UTF-8-Zeichen werden als ASCII-Zeichen behandelt.

- Anfragen sind erfolgreich, wenn die benutzerdefinierten Metadaten maskierte UTF-8-Zeichen enthalten.
- StorageGRID gibt den `x-amz-missing-meta` Header nicht zurück, wenn der interpretierte Wert des Schlüsselnamens oder -werts nicht druckbare Zeichen enthält.

Unterstützte Anfrageheader

Die folgenden Anfrage-Header 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 Namen-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 `REPLACE` angeben, 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 `REPLACE` angeben, 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`

    Wird eine Anfrage ohne diese Header gestellt, werden die Standard-Aufbewahrungseinstellungen des Buckets verwendet, um den Objektversionsmodus und das Aufbewahrungsdatum zu berechnen. Siehe "S3 REST API zur Konfiguration von S3 Object Lock verwenden".
- 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`

  Siehe Anfrage-Header für serverseitige Verschlüsselung

Nicht unterstützte Anfrageheader

Die folgenden Anfrageheader werden nicht unterstützt:

- `Cache-Control`
- `Content-Disposition`
- `Content-Encoding`
- `Content-Language`
- `Expires`
- `If-Match`

  Das `If-Match header` wird akzeptiert, ist aber nicht funktionsfähig.

- `If-None-Match`

  Das `If-None-Match header` wird akzeptiert, ist aber nicht funktionsfähig.

- `x-amz-checksum-algorithm`

  Wenn ein Objekt kopiert wird, wird der Prüfsummenwert des Quellobjekts von StorageGRID nicht auf das neue Objekt übertragen, selbst wenn das Quellobjekt eine Prüfsumme hat. Dieses Verhalten gilt unabhängig davon, ob `x-amz-checksum-algorithm` in der Objektanforderung verwendet wird.

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

Speicherklassenoptionen

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

- `STANDARD`

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

- `REDUCED_REDUNDANCY`

  Gibt einen Einzel-Commit-Aufnahmevorgang an, wenn die ILM-Regel die Dual commit Option verwendet oder wenn die Balanced Option auf die Erstellung von Zwischenkopien zurückgreift.

Wenn Sie ein Objekt in einen Bucket mit aktiviertem S3 Object Lock importieren, wird die `REDUCED_REDUNDANCY` Option ignoriert. Wenn Sie ein Objekt in einen älteren Compliant-Bucket importieren, gibt die `REDUCED_REDUNDANCY` Option einen Fehler zurück. StorageGRID führt immer einen Dual-Commit-Import durch, um die Einhaltung der Compliance-Anforderungen zu gewährleisten.

Verwendung von x-amz-copy-source in CopyObject

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

Stimmen Quelle und Ziel überein und ist der `x-amz-metadata-directive` Header als `REPLACE` angegeben, werden die Metadaten des Objekts mit den in der Anfrage übermittelten Metadatenwerten aktualisiert. In diesem Fall wird das Objekt von StorageGRID nicht erneut eingelesen. Dies hat zwei wichtige Konsequenzen:

- Sie können CopyObject nicht verwenden, um ein bestehendes Objekt direkt zu verschlüsseln oder die Verschlüsselung eines bestehenden Objekts direkt zu ändern. Wenn Sie den `x-amz-server-side-encryption`-Header oder den `x-amz-server-side-encryption-customer-algorithm`-Header angeben, lehnt StorageGRID die Anfrage ab und gibt `XNotImplemented` zurück.
- Die in der entsprechenden ILM-Regel festgelegte Option für das Aufnahmeverhalten wird nicht verwendet. Alle durch das Update ausgelösten Änderungen an der Objektplatzierung erfolgen, wenn ILM durch normale Hintergrundprozesse erneut ausgewertet wird.

  Das bedeutet, dass bei Verwendung der Option „Streng“ für das Aufnahmeverhalten in der ILM-Regel keine Aktion ausgeführt wird, wenn die erforderlichen Objektplatzierungen nicht möglich sind (zum Beispiel, weil ein neu benötigter Speicherort nicht verfügbar ist). Das aktualisierte Objekt behält seine aktuelle Platzierung bei, bis die erforderliche Platzierung möglich ist.

Anfrage-Header für serverseitige Verschlüsselung

Wenn Sie "serverseitige Verschlüsselung" dies tun, hängen die von Ihnen angegebenen Anfrageheader davon ab, ob das Quellobjekt verschlüsselt ist und ob Sie planen, das Zielobjekt zu verschlüsseln.

- Wenn das Quellobjekt mit einem vom Kunden bereitgestellten Schlüssel (SSE-C) verschlüsselt ist, müssen die folgenden drei Header in die CopyObject-Anfrage aufgenommen werden, damit das Objekt entschlüsselt und anschließend kopiert werden kann:
  - `x-amz-copy-source-server-side-encryption-customer-algorithm`: Geben Sie `AES256` an.
  - `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 von Ihnen bereitgestellten und verwalteten eindeutigen Schlüssel verschlüsseln möchten, sind die folgenden drei Header einzuschließen:
  - `x-amz-server-side-encryption-customer-algorithm`: Angeben `AES256`.
  - `x-amz-server-side-encryption-customer-key`: Ein neuer Verschlüsselungsschlüssel für das Zielobjekt wird angegeben.

- `x-amz-server-side-encryption-customer-key-MD5`: Geben Sie den MD5-Hash des neuen Verschlüsselungsschlüssels an.

> Die von Ihnen bereitgestellten Verschlüsselungsschlüssel werden niemals gespeichert. Wenn Sie einen Verschlüsselungsschlüssel verlieren, verlieren Sie das zugehörige Objekt. Vor der Verwendung von kundenseitig bereitgestellten Schlüsseln zur Sicherung von Objektdaten sollten die Hinweise zu "Verwendung serverseitiger Verschlüsselung" beachtet werden.

- Wenn Sie das Zielobjekt (die Kopie) mit einem von StorageGRID (SSE) verwalteten eindeutigen Schlüssel verschlüsseln möchten, fügen Sie diesen Header in die CopyObject-Anfrage ein:
  - `x-amz-server-side-encryption`

> Der `server-side-encryption` Wert des Objekts kann nicht aktualisiert werden. Stattdessen kann eine Kopie mit einem neuen `server-side-encryption` Wert unter Verwendung von `x-amz-metadata-directive`: `REPLACE` erstellt werden.

Versionierung

Wenn der Quell-Bucket versioniert ist, kann der `x-amz-copy-source` Header verwendet werden, um die neueste Version eines Objekts zu kopieren. Um eine bestimmte Version eines Objekts zu kopieren, muss die zu kopierende Version explizit mithilfe der `versionId` Subressource angegeben werden. Wenn der Ziel-Bucket versioniert ist, wird die generierte Version im `x-amz-version-id` Antwortheader zurückgegeben. Wenn die Versionierung für den Ziel-Bucket ausgesetzt ist, gibt `x-amz-version-id` einen „null“-Wert zurück.

Rufen Sie ein Objekt aus einem Bucket in StorageGRID mit der GetObject-Anfrage ab

Die S3 GetObject-Anfrage kann verwendet werden, um ein Objekt aus einem S3-Bucket abzurufen.

GetObject und mehrteilige Objekte

Der `partNumber`Anfrageparameter kann verwendet werden, um einen bestimmten Teil eines mehrteiligen oder segmentierten Objekts abzurufen. Das `x-amz-mp-parts-count`Antwortelement gibt an, wie viele Teile das Objekt hat.

Sie können `partNumber` auf 1 setzen, sowohl für segmentierte/multipart Objekte als auch für nicht segmentierte/nicht multipart Objekte; das `x-amz-mp-parts-count` Antwortelement wird jedoch nur für segmentierte oder multipart Objekte zurückgegeben.

UTF-8-Zeichen in Benutzermetadaten

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

Unterstützter Anfrageheader

Der folgende Anfrageheader wird unterstützt:

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

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

Nicht unterstützter Anfrageheader

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

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

Versionierung

Wenn keine `versionId` Unterressource angegeben ist, wird mit dem Vorgang die neueste Version des Objekts in einem versionierten Bucket abgerufen. Wenn die aktuelle Version des Objekts eine Löschmarkierung ist, wird ein „Nicht gefunden“-Status zurückgegeben, wobei der `x-amz-delete-marker` Antwortheader auf `true` gesetzt wird.

Anfrage-Header für serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C)

Alle drei Header sind zu verwenden, 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 die MD5-Prüfsumme des Verschlüsselungsschlüssels des Objekts an.

> Die von Ihnen bereitgestellten Verschlüsselungsschlüssel werden niemals gespeichert. Wenn ein Verschlüsselungsschlüssel verloren geht, geht auch das zugehörige Objekt verloren. Vor der Verwendung von kundenseitig bereitgestellten Schlüsseln zur Sicherung von Objektdaten sind die Hinweise in "Serverseitige Verschlüsselung verwenden" zu beachten.

Verhalten von GetObject für Cloud Storage Pool-Objekte

Wenn ein Objekt in einem "Cloud Storage Pool" gespeichert wurde, hängt das Verhalten einer GetObject-Anfrage vom Zustand des Objekts ab. Siehe "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, wird bei GetObject-Anfragen versucht, die Daten zuerst aus dem Grid abzurufen, bevor sie aus dem Cloud-Speicherpool abgerufen werden.

Zustand des Objekts	Verhalten von GetObject
Objekt, das in StorageGRID aufgenommen, aber noch nicht von ILM ausgewertet wurde, oder Objekt, das in einem herkömmlichen Speicherpool gespeichert oder mit Erasure Coding abgelegt ist	`200 OK` Eine Kopie des Objekts wird abgerufen.

Zustand des Objekts	Verhalten von GetObject
Objekt im Cloud Storage Pool, aber noch nicht in einen nicht abrufbaren Zustand überführt	`200 OK` Eine Kopie des Objekts wird abgerufen.
Das Objekt ist in einen nicht abrufbaren Zustand übergegangen.	`403 Forbidden`, `InvalidObjectState` Eine "RestoreObject"-Anfrage kann verwendet werden, um das Objekt in einen abrufbaren Zustand zurückzuversetzen.
Objekt befindet sich im Wiederherstellungsprozess aus einem nicht wiederherstellbaren Zustand	`403 Forbidden`, `InvalidObjectState` Warten, bis die RestoreObject-Anfrage abgeschlossen ist.
Objekt vollständig im Cloud Storage Pool wiederhergestellt	`200 OK` Eine Kopie des Objekts wird abgerufen.

Mehrteilige oder segmentierte Objekte in einem 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 eine Teilmenge der Objektteile oder -segmente geprüft wird. In einigen Fällen kann eine GetObject-Anfrage fälschlicherweise `200 OK` zurückgeben, wenn einige Teile des Objekts bereits in einen nicht wiederherstellbaren Zustand überführt wurden oder wenn einige Teile des Objekts noch nicht wiederhergestellt wurden.

In diesen Fällen:

- Die GetObject Anfrage liefert möglicherweise einige Daten zurück, bricht aber mitten in der Übertragung ab.
- Eine nachfolgende GetObject-Anfrage könnte `403 Forbidden` zurückgeben.

GetObject und gridübergreifende Replikation

Wenn Sie "Grid-Föderation" verwenden und "gridübergreifende Replikation" für einen Bucket aktiviert ist, kann der S3-Client den Replikationsstatus eines Objekts durch eine GetObject-Anfrage überprüfen. Die Antwort enthält den StorageGRID-spezifischen `x-ntap-sg-cgr-replication-status` Response-Header, der einen der folgenden Werte aufweist:

Grid	Replikationsstatus
Quelle	• **ABGESCHLOSSEN**: Die Replikation war erfolgreich. • **AUSSTEHEND**: Das Objekt ist noch nicht repliziert worden. • **FEHLER**: Die Replikation ist dauerhaft fehlgeschlagen. Ein Benutzer muss den Fehler beheben.

Grid	Replikationsstatus
Ziel	**REPLICA**: Das Objekt wurde aus dem Quellgrid repliziert.

StorageGRID unterstützt den `x-amz-replication-status` Header nicht.

Metadaten eines Objekts in StorageGRID mit der S3 HeadObject-Anfrage abrufen

Sie können die S3 HeadObject Anfrage verwenden, um Metadaten eines Objekts abzurufen, ohne das Objekt selbst zurückzugeben. Wenn das Objekt in einem Cloud Storage Pool gespeichert ist, kann HeadObject verwendet werden, um den Übergangsstatus des Objekts zu ermitteln.

HeadObject und mehrteilige Objekte

Mithilfe des `partNumber` Anfrageparameters können Metadaten für einen bestimmten Teil eines mehrteiligen oder segmentierten Objekts abgerufen werden. Das `x-amz-mp-parts-count` Antwortelement gibt an, aus wie vielen Teilen das Objekt besteht.

Sie können `partNumber` auf 1 setzen, sowohl für segmentierte/multipart Objekte als auch für nicht segmentierte/nicht multipart Objekte; das `x-amz-mp-parts-count` Antwortelement wird jedoch nur für segmentierte oder multipart Objekte zurückgegeben.

UTF-8-Zeichen in Benutzermetadaten

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

Unterstützter Anfrageheader

Der folgende Anfrageheader wird unterstützt:

- `x-amz-checksum-mode`

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

Nicht unterstützter Anfrageheader

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

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

Versionierung

Wenn keine `versionId` Unterressource angegeben ist, wird mit dem Vorgang die neueste Version des Objekts in einem versionierten Bucket abgerufen. Wenn die aktuelle Version des Objekts eine Löschmarkierung ist, wird ein „Nicht gefunden“-Status zurückgegeben, wobei der `x-amz-delete-marker`

Antwortheader auf `true` gesetzt wird.

Anfrage-Header für serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C)

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

- `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 die MD5-Prüfsumme des Verschlüsselungsschlüssels des Objekts an.

> Die von Ihnen bereitgestellten Verschlüsselungsschlüssel werden niemals gespeichert. Wenn ein Verschlüsselungsschlüssel verloren geht, geht auch das zugehörige Objekt verloren. Vor der Verwendung von kundenseitig bereitgestellten Schlüsseln zur Sicherung von Objektdaten sind die Hinweise in "Serverseitige Verschlüsselung verwenden" zu beachten.

HeadObject-Antworten für Cloud Storage Pool-Objekte

Wenn das Objekt in einem "Cloud Storage Pool" gespeichert ist, werden die folgenden Antwortheader zurückgegeben:

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

Die Antwortheader liefern Informationen über den Zustand eines Objekts, wenn es in einen Cloud Storage Pool verschoben, optional in einen nicht abrufbaren Zustand überführt und wiederhergestellt wird.

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

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

Zustand des Objekts	Antwort auf HeadObject

Das Objekt ist in einen nicht abrufbaren Zustand übergegangen, aber mindestens eine Kopie existiert noch im Grid.	`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` wird auf einen weit entfernten Zeitpunkt in der Zukunft festgelegt.
	
	Hinweis: Wenn die Kopie im Grid nicht verfügbar ist (zum Beispiel weil ein Storage Node ausgefallen ist), muss eine "RestoreObject"-Anfrage zur Wiederherstellung der Kopie aus dem Cloud Storage Pool gestellt werden, bevor das Objekt erfolgreich abgerufen werden kann.

Das Objekt ist in einen nicht wiederherstellbaren Zustand übergegangen, und es existiert keine Kopie im Grid.	`200 OK`
	
	`x-amz-storage-class: GLACIER`

Objekt befindet sich im Wiederherstellungsprozess aus einem nicht wiederherstellbaren Zustand	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="true"`

Objekt vollständig im Cloud Storage Pool 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"`
	
	Das `expiry-date`zeigt an, wann sich das Objekt im Cloud Storage Pool wieder in einem nicht abrufbaren Zustand befindet.

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 eine Teilmenge der Objektteile oder -segmente abgefragt wird. In einigen Fällen kann eine HeadObject-Anfrage fälschlicherweise `x-amz-restore: ongoing-request="false"` zurückgeben, wenn einige Teile des Objekts bereits in einen nicht wiederherstellbaren Zustand überführt wurden oder wenn einige Teile des Objekts noch nicht wiederhergestellt wurden.

HeadObject und gridübergreifende Replikation

Wenn Sie "Grid-Föderation" verwenden und "gridübergreifende Replikation" für einen Bucket aktiviert ist, kann der S3-Client den Replikationsstatus eines Objekts durch eine HeadObject-Anfrage überprüfen. Die Antwort enthält den StorageGRID-spezifischen `x-ntap-sg-cgr-replication-status` Response-Header, der einen der folgenden Werte aufweist:

Grid	Replikationsstatus
Quelle	• **ABGESCHLOSSEN**: Die Replikation war erfolgreich. • **AUSSTEHEND**: Das Objekt ist noch nicht repliziert worden. • **FEHLER**: Die Replikation ist dauerhaft fehlgeschlagen. Ein Benutzer muss den Fehler beheben.
Ziel	**REPLICA**: Das Objekt wurde aus dem Quellgrid repliziert.

StorageGRID unterstützt den `x-amz-replication-status` Header nicht.

Fügen Sie mit der S3 PutObject-Anfrage ein Objekt zu einem Bucket in StorageGRID hinzu

Sie können die S3 PutObject Anfrage verwenden, um ein Objekt zu einem Bucket hinzuzufügen.

Konflikte beheben

Konfliktierende Clientanfragen, beispielsweise wenn zwei Clients auf denselben Schlüssel schreiben, werden nach dem Prinzip „latest-wins“ aufgelöst. Der Zeitpunkt der „latest-wins“-Auswertung richtet sich danach, wann das StorageGRID System die jeweilige Anfrage abgeschlossen hat, und nicht danach, wann S3 Clients eine Operation starten.

Objektgröße

Die maximal empfohlene Größe für einen einzelnen PutObject-Vorgang beträgt 5 GiB (5.368.709.120 Byte). Bei Objekten, die größer als 5 GiB sind, sollte stattdessen "mehrteiliger Upload" verwendet werden.

Die maximal *unterstützte* Größe für eine einzelne PutObject-Operation beträgt 5 TiB (5.497.558.138.880 Bytes).

Wenn Sie von StorageGRID 11.6 oder einer älteren Version aktualisiert haben, wird die Warnung „S3 PUT Object size too large“ ausgelöst, wenn Sie versuchen, ein Objekt mit mehr als 5 GiB hochzuladen. Bei einer Neuinstallation von StorageGRID 11.7 oder 11.8 wird die Warnung in diesem Fall nicht ausgelöst. Um jedoch dem AWS S3-Standard zu entsprechen, werden zukünftige Versionen von StorageGRID keine Uploads von Objekten mit mehr als 5 GiB unterstützen.

Größe der Benutzermetadaten

Amazon S3 beschränkt die Größe benutzerdefinierter Metadaten in jedem PUT-Anfrageheader auf 2 KB. StorageGRID beschränkt benutzerdefinierte Metadaten auf 24 KiB. Die Größe benutzerdefinierter Metadaten wird durch die Summe der Anzahl der Bytes in der UTF-8-Kodierung jedes Schlüssels und Werts gemessen.

UTF-8-Zeichen in Benutzermetadaten

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

StorageGRID analysiert oder interpretiert keine maskierten UTF-8-Zeichen, die im Schlüsselname oder -wert benutzerdefinierter Metadaten enthalten sind. Maskierte UTF-8-Zeichen werden als ASCII-Zeichen behandelt.

- PutObject, CopyObject, GetObject und HeadObject-Anfragen sind erfolgreich, wenn die benutzerdefinierten Metadaten maskierte UTF-8-Zeichen enthalten.
- StorageGRID gibt den `x-amz-missing-meta` Header nicht zurück, 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 bestehenden Objekten hinzufügen. Sowohl StorageGRID als auch Amazon S3 unterstützen bis zu 10 Tags pro Objekt. Tags, die einem Objekt zugeordnet sind, 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. Schlüssel und Werte sind Groß-/Kleinschreibung.

Objektbesitz

In StorageGRID gehören alle Objekte dem Bucket-Besitzerkonto, einschließlich Objekten, die von einem Nicht-Besitzerkonto oder einem anonymen Benutzer erstellt wurden.

Unterstützte Anfrageheader

Die folgenden Anfrage-Header werden unterstützt:

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

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

  - StorageGRID überprüft die `chunk-signature` nicht anhand der Chunk-Daten.
  - StorageGRID überprüft den von Ihnen angegebenen Wert `x-amz-decoded-content-length`nicht anhand des Objekts.
- `Content-Language`
- `Content-Length`
- `Content-MD5`
- `Content-Type`
- `Expires`
- `Transfer-Encoding`

  Chunked Transfer-Codierung wird unterstützt, wenn `aws-chunked` Nutzlast-Signierung ebenfalls verwendet wird.

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

  Beim Festlegen des Namen-Wert-Paares für benutzerdefinierte Metadaten wird dieses allgemeine Format verwendet:

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

  Wenn die Option **Benutzerdefinierte Erstellungszeit** als Referenzzeit für eine ILM-Regel verwendet werden soll, muss `creation-time` als Name der Metadaten verwendet werden, die den Erstellungszeitpunkt des Objekts aufzeichnen. Beispielsweise:

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

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

  > Eine ILM-Regel kann nicht sowohl eine **benutzerdefinierte Erstellungszeit** für die Referenzzeit als auch die Aufnahmeoption Balanced oder Strict verwenden. Beim Erstellen der ILM-Regel tritt ein Fehler auf.

- `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`

    Wird eine Anfrage ohne diese Header gestellt, werden die Standard-Aufbewahrungseinstellungen des Buckets verwendet, um den Objektversionsmodus und das Aufbewahrungsdatum zu berechnen. Siehe "S3 REST API zur Konfiguration von S3 Object Lock verwenden".

- 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`

    Siehe Anfrage-Header für serverseitige Verschlüsselung

Nicht unterstützte Anfrageheader

Die folgenden Anfrageheader werden nicht unterstützt:

- `If-Match`

Das `If-Match header` wird akzeptiert, ist aber nicht funktionsfähig.

- `If-None-Match`

  Das `If-None-Match header` wird akzeptiert, ist aber nicht funktionsfähig.

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

  Der `x-amz-website-redirect-location`Header gibt `XNotImplemented` zurück.

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 (dies wird durch ILM bestimmt).

Wenn die ILM-Regel, die auf ein aufgenommenes Objekt zutrifft, die Strict-Aufnahmeoption verwendet, hat der `x-amz-storage-class` Header keine Auswirkung.

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

- `STANDARD` (Standard)
  - **Doppelter Commit**: Wenn die ILM-Regel die Option „Doppelter Commit“ für das Aufnahmeverhalten festlegt, wird nach der Aufnahme eines Objekts eine zweite Kopie dieses Objekts erstellt und auf einem anderen Storage Node verteilt (doppelter Commit). Bei der Auswertung der ILM wird von StorageGRID geprüft, ob diese anfänglichen Zwischenkopien den Platzierungsanweisungen in der Regel entsprechen. Falls dies nicht der Fall ist, müssen möglicherweise neue Objektkopien an anderen Speicherorten erstellt und die anfänglichen Zwischenkopien gelöscht werden.
  - **Ausgeglichen**: Wenn die ILM-Regel die Option „Ausgeglichen“ vorsieht und StorageGRID nicht sofort alle in der Regel angegebenen Kopien erstellen kann, erstellt StorageGRID zwei Zwischenkopien auf verschiedenen Storage Nodes.

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

- `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 Auswirkung. Die `REDUCED_REDUNDANCY` Option ist am besten geeignet, wenn die ILM-Regel, die auf das Objekt zutrifft, eine einzelne replizierte Kopie erstellt. In diesem Fall wird durch die Verwendung von `REDUCED_REDUNDANCY` diese Option das unnötige Erstellen und Löschen einer zusätzlichen Objektkopie bei jedem Ingest-Vorgang vermieden.

Die Verwendung der `REDUCED_REDUNDANCY` Option wird unter anderen Umständen nicht empfohlen. `REDUCED_REDUNDANCY` Das Risiko von Objekt-Datenverlust während der Aufnahme wird dadurch erhöht. Beispielsweise kann es zu Datenverlust kommen, wenn die einzige Kopie zunächst auf einem Storage Node gespeichert wird, der ausfällt, bevor die ILM-Auswertung erfolgen kann.

> Wenn für einen bestimmten Zeitraum nur eine replizierte Kopie vorhanden ist, besteht das Risiko eines dauerhaften Datenverlusts. Wenn nur eine replizierte Kopie eines Objekts existiert, geht dieses Objekt verloren, wenn ein Storage Node ausfällt oder einen schwerwiegenden Fehler aufweist. Auch während Wartungsarbeiten wie Upgrades ist der Zugriff auf das Objekt vorübergehend nicht möglich.

Die Angabe von `REDUCED_REDUNDANCY` wirkt sich lediglich darauf aus, wie viele Kopien beim erstmaligen Einlesen eines Objekts erstellt werden. Sie hat keinen Einfluss darauf, wie viele Kopien des Objekts bei der Auswertung durch die aktiven ILM-Richtlinien erstellt werden und führt nicht dazu, dass Daten im StorageGRID System mit geringerer Redundanz gespeichert werden.

> Wenn Sie ein Objekt in einen Bucket mit aktiviertem S3 Object Lock importieren, wird die `REDUCED_REDUNDANCY` Option ignoriert. Wenn Sie ein Objekt in einen älteren Compliant-Bucket importieren, gibt die `REDUCED_REDUNDANCY` Option einen Fehler zurück. StorageGRID führt immer einen Dual-Commit-Import durch, um die Einhaltung der Compliance-Anforderungen zu gewährleisten.

Anfrage-Header für serverseitige Verschlüsselung

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

- **SSE**: Der folgende Header ist zu verwenden, wenn das Objekt mit einem von StorageGRID verwalteten eindeutigen Schlüssel verschlüsselt werden soll.
  - `x-amz-server-side-encryption`

    Wenn der `x-amz-server-side-encryption`Header nicht in der PutObject-Anfrage enthalten ist, wird die grid-weite "Einstellung für die Verschlüsselung gespeicherter Objekte" aus der PutObject-Antwort ausgelassen.

- **SSE-C**: Alle drei dieser Header werden verwendet, wenn das Objekt mit einem eindeutigen Schlüssel verschlüsselt werden soll, 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`: Der MD5-Digest des Verschlüsselungsschlüssels des neuen Objekts ist anzugeben.

> Die von Ihnen bereitgestellten Verschlüsselungsschlüssel werden niemals gespeichert. Wenn Sie einen Verschlüsselungsschlüssel verlieren, verlieren Sie das zugehörige Objekt. Vor der Verwendung von kundenseitig bereitgestellten Schlüsseln zur Sicherung von Objektdaten sollten die Hinweise zu "Verwendung serverseitiger Verschlüsselung" beachtet werden.

> 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, wird automatisch eine eindeutige `versionId` für die Version des gespeicherten Objekts generiert. Diese `versionId` wird auch in der Antwort mit dem `x-amz-version-id` Response-Header zurückgegeben.

Wenn die Versionsverwaltung ausgesetzt ist, wird die Objektversion mit einem Nullwert `versionId` gespeichert, und falls bereits eine Nullversion existiert, wird diese überschrieben.

Signaturberechnungen für den Authorization-Header

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

- StorageGRID erfordert nicht, dass `host`-Header innerhalb von `CanonicalHeaders` enthalten sind.
- StorageGRID erfordert nicht, dass `Content-Type` innerhalb von `CanonicalHeaders` enthalten ist.
- StorageGRID erfordert nicht, dass `x-amz-*`-Header innerhalb von `CanonicalHeaders` enthalten sind.

> Als allgemeine Best Practice sollten diese Header immer innerhalb von `CanonicalHeaders` enthalten sein, damit sie überprüft werden; wenn diese Header jedoch ausgeschlossen werden, gibt StorageGRID keinen Fehler zurück.

Weitere Informationen finden Sie unter "Signaturberechnungen für den Authorization Header: Übertragung der Nutzlast in einem einzigen Chunk (AWS Signature Version 4)".

Verwandte Informationen

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

Ein Objekt in StorageGRID mit der S3-RestoreObject-Anfrage wiederherstellen

Sie können die S3 RestoreObject Anfrage verwenden, um ein Objekt wiederherzustellen, das in einem Cloud Storage Pool gespeichert ist.

Unterstützter Anfragetyp

StorageGRID unterstützt ausschließlich RestoreObject-Anfragen zur Wiederherstellung eines Objekts. Die `SELECT` Art der Wiederherstellung wird nicht unterstützt. Select-Anfragen geben `XNotImplemented` zurück.

Versionierung

Optional kann `versionId` angegeben werden, um eine bestimmte Version eines Objekts in einem versionierten Bucket wiederherzustellen. Wenn `versionId` nicht angegeben wird, wird die neueste Version des Objekts wiederhergestellt.

Verhalten von RestoreObject auf Cloud Storage Pool Objekten

Wenn ein Objekt in einem "Cloud Storage Pool" gespeichert wurde, weist eine RestoreObject-Anfrage abhängig vom Zustand des Objekts folgendes Verhalten auf. Weitere Einzelheiten sind unter "HeadObject" zu finden.

Wenn ein Objekt in einem Cloud Storage Pool gespeichert ist und eine oder mehrere Kopien des Objekts auch im Grid vorhanden sind, ist es nicht erforderlich, das Objekt durch eine RestoreObject Anfrage wiederherzustellen. Stattdessen kann die lokale Kopie direkt mit einer GetObject Anfrage 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 Storage Pool	`403 Forbidden`, `InvalidObjectState`
Objekt im Cloud Storage Pool, aber noch nicht in einen nicht abrufbaren Zustand überführt	`200 OK`Es werden keine Änderungen vorgenommen. **Hinweis**: Bevor ein Objekt in einen nicht abrufbaren Zustand überführt wurde, können Sie dessen `expiry-date` nicht ändern.
Das Objekt ist in einen nicht abrufbaren Zustand übergegangen.	`202 Accepted`Stellt eine abrufbare Kopie des Objekts für die im Anfragetext angegebene Anzahl von Tagen im Cloud Storage Pool wieder her. Nach Ablauf dieses Zeitraums befindet sich das Objekt wieder in einem nicht abrufbaren Zustand. Optional kann das `Tier`Anforderungselement verwendet werden, um zu bestimmen, wie lange der Wiederherstellungsauftrag bis zum Abschluss benötigt (`Expedited`, `Standard` oder `Bulk`). Wenn `Tier` nicht angegeben wird, wird die `Standard` Tier verwendet. **Wichtig**: Wenn ein Objekt in S3 Glacier Deep Archive verschoben wurde oder der Cloud Storage Pool Azure Blob Storage verwendet, kann es nicht mit der `Expedited`-Ebene wiederhergestellt werden. Der folgende Fehler wird zurückgegeben `403 Forbidden`, `InvalidTier`: `Retrieval option is not supported by this storage class`.
Objekt befindet sich im Wiederherstellungsprozess aus einem nicht wiederherstellbaren Zustand	`409 Conflict`, `RestoreAlreadyInProgress`
Objekt vollständig im Cloud Storage Pool wiederhergestellt	`200 OK` **Hinweis:** Wenn ein Objekt in einen abrufbaren Zustand wiederhergestellt wurde, kann sein `expiry-date` durch erneutes Senden der RestoreObject-Anfrage mit einem neuen Wert für `Days` geändert werden. Das Wiederherstellungsdatum wird relativ zum Zeitpunkt der Anfrage aktualisiert.

Filtern Sie Objektdaten in StorageGRID mit der S3 SelectObjectContent-Anfrage

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

Weitere Informationen siehe "Amazon Simple Storage Service API-Referenz: SelectObjectContent".

Bevor Sie beginnen

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

Beispiel für die Syntax einer CSV-Anfrage

```
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 Syntax einer Parquet-Anfrage

```
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 aus den US-Volkszählungsdaten. Datensätze in der Datei, die keine Bundesstaaten 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 folgendermaßen 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 Zeilen der Ausgabedatei, `changes.csv` sehen folgendermaßen 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 Anwendungsbeispiel (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
```

Operationen für mehrteilige Uploads

Unterstützte mehrteilige Upload-Operationen in StorageGRID

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

Für alle mehrteiligen Upload-Vorgänge gelten die folgenden Bedingungen und Hinweise:

- Sie sollten nicht mehr als 1.000 gleichzeitige Multipart-Uploads in einen einzelnen Bucket durchführen, da die Ergebnisse von ListMultipartUploads-Abfragen für diesen Bucket möglicherweise unvollständig sind.
- StorageGRID setzt die AWS-Größenbeschränkungen für Multipart-Parts durch. S3-Clients müssen diese Richtlinien beachten:
  - 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 Teilgrößen so groß wie möglich sein. Beispielsweise eignen sich Teilgrößen von 5 GiB für ein 100 GiB Objekt. Da jeder Teil als separates Objekt betrachtet wird, verringert die Verwendung großer Teilgrößen den Metadaten-Overhead von StorageGRID.
  - Für Objekte, die kleiner als 5 GiB sind, empfiehlt sich stattdessen die Verwendung eines Uploads ohne mehrere Teile.
- ILM wird für jeden Teil eines mehrteiligen Objekts beim Einlesen und für das gesamte Objekt nach Abschluss des mehrteiligen Uploads ausgewertet, wenn die ILM-Regel den Modus Ausgewogen oder Streng "Aufnahmeoption" verwendet. Es ist zu berücksichtigen, wie sich dies auf die Platzierung von Objekten und Teilen auswirkt:
  - Wenn sich ILM während eines S3-Multipart-Uploads ändert, erfüllen einige Teile des Objekts beim

Abschluss des Multipart-Uploads möglicherweise nicht die aktuellen ILM-Anforderungen. Jeder Teil, der nicht korrekt platziert ist, wird zur erneuten ILM-Bewertung in die Warteschlange gestellt und später an den richtigen Speicherort verschoben.

  - Bei der Auswertung der ILM-Kriterien für ein Teil filtert StorageGRID anhand der Größe des Teils, nicht anhand der Größe des Objekts. Das bedeutet, dass Teile eines Objekts an Standorten gespeichert werden können, die die ILM-Anforderungen für das Objekt als Ganzes nicht erfüllen. Wenn beispielsweise eine Regel festlegt, dass alle Objekte ab 10 GB in DC1 gespeichert werden, während alle kleineren Objekte in DC2 gespeichert werden, wird jedes 1-GB-Teil eines 10-teiligen Multipart-Uploads beim Import in DC2 gespeichert. Wenn jedoch die ILM-Kriterien für das Objekt als Ganzes ausgewertet werden, werden alle Teile des Objekts nach DC1 verschoben.

- Alle Multipart-Upload-Operationen unterstützen StorageGRID "Konsistenzwerte".
- Wenn ein Objekt mit Multipart-Upload importiert wird, wird "Objektsegmentierungsschwelle (1 GiB)" nicht angewendet.
- Bei Bedarf kann "serverseitige Verschlüsselung" mit Multipart-Uploads verwendet werden. Für die Nutzung von SSE (serverseitige Verschlüsselung mit von StorageGRID verwalteten Schlüsseln) ist der `x-amz-server-side-encryption` Request-Header ausschließlich in der CreateMultipartUpload-Anfrage einzuschließen. Für die Nutzung von SSE-C (serverseitige Verschlüsselung mit kundenseitig bereitgestellten Schlüsseln) sind dieselben drei Verschlüsselungsschlüssel-Request-Header sowohl in der CreateMultipartUpload-Anfrage als auch in jeder nachfolgenden UploadPart-Anfrage anzugeben.
- Ein mehrteilig hochgeladenes Objekt ist in einer "Branch-Bucket" enthalten, wenn die Aufnahme vor dem Before-Zeitstempel des Basis-Buckets initiiert wurde, unabhängig davon, wann der Upload abgeschlossen wird.

Betrieb	Implementierung
AbortMultipartUpload	Implementiert mit dem gesamten Verhalten der Amazon S3 REST API. Änderungen vorbehalten.
CompleteMultipartUpload	Siehe "CompleteMultipartUpload"
CreateMultipartUpload (früher Initiate Multipart Upload genannt)	Siehe "CreateMultipartUpload"
ListMultipartUploads	Siehe "ListMultipartUploads"
ListParts	Implementiert mit dem gesamten Verhalten der Amazon S3 REST API. Änderungen vorbehalten.
UploadPart	Siehe "UploadPart"
UploadPartCopy	Siehe "UploadPartCopy"

Wie StorageGRID S3 REST API mehrteilige Uploads durchführt

Der CompleteMultipartUpload-Vorgang schließt einen mehrteiligen Upload eines Objekts ab, indem die zuvor hochgeladenen Teile zusammengefügt werden.

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

Konflikte beheben

Konfliktierende Clientanfragen, beispielsweise wenn zwei Clients auf denselben Schlüssel schreiben, werden nach dem Prinzip „latest-wins“ aufgelöst. Der Zeitpunkt der „latest-wins“-Auswertung richtet sich danach, wann das StorageGRID System die jeweilige Anfrage abgeschlossen hat, und nicht danach, wann S3 Clients eine Operation starten.

Unterstützte Anfrageheader

Die folgenden Anfrage-Header werden unterstützt:

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

  Der `x-amz-storage-class`Header beeinflusst, wie viele Objektkopien StorageGRID erstellt, wenn die entsprechende ILM-Regel die "Dual Commit oder Balanced Ingest Option" angibt.

- `STANDARD`

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

- `REDUCED_REDUNDANCY`

  Gibt einen Einzel-Commit-Aufnahmevorgang an, wenn die ILM-Regel die Dual commit Option verwendet oder wenn die Balanced Option auf die Erstellung von Zwischenkopien zurückgreift.

  Wenn Sie ein Objekt in einen Bucket mit aktiviertem S3 Object Lock importieren, wird die `REDUCED_REDUNDANCY` Option ignoriert. Wenn Sie ein Objekt in einen älteren Compliant-Bucket importieren, gibt die `REDUCED_REDUNDANCY` Option einen Fehler zurück. StorageGRID führt immer einen Dual-Commit-Import durch, um die Einhaltung der Compliance-Anforderungen zu gewährleisten.

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

Der `ETag` Wert, der zurückgegeben wird, ist keine MD5-Summe der Daten, sondern entspricht der Amazon S3 API-Implementierung des `ETag` Werts für Multipart-Objekte.

Nicht unterstützte Anfrageheader

Die folgenden Anfrageheader werden nicht unterstützt:

- `If-Match`

  Das `If-Match header` wird akzeptiert, ist aber nicht funktionsfähig.

- `If-None-Match`

  Das `If-None-Match header` wird akzeptiert, ist aber nicht funktionsfähig.

- `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, wird automatisch eine eindeutige `versionId` für die Version des gespeicherten Objekts generiert. Diese `versionId` wird auch in der Antwort mit dem `x-amz-version-id` Response-Header zurückgegeben.

Wenn die Versionsverwaltung ausgesetzt ist, wird die Objektversion mit einem Nullwert `versionId` gespeichert, und falls bereits eine Nullversion existiert, wird diese überschrieben.

> Wenn die Versionierung für einen Bucket aktiviert ist, wird bei jedem Multipart-Upload immer eine neue Version erstellt, selbst wenn mehrere Multipart-Uploads gleichzeitig für denselben Objektschlüssel abgeschlossen werden. Wenn die Versionierung für einen Bucket nicht aktiviert ist, kann ein Multipart-Upload initiiert werden und anschließend ein weiterer Multipart-Upload für denselben Objektschlüssel zuerst initiiert und abgeschlossen werden. Bei Buckets ohne Versionierung hat der Multipart-Upload, der zuletzt abgeschlossen wird, Vorrang.

Fehlgeschlagene Replikation, Benachrichtigung oder Metadatenbenachrichtigung

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

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

Weitere Informationen finden Sie unter "Fehlerbehebung bei Plattformdiensten".

S3 CreateMultipartUpload Anforderungsheader und Optionen in StorageGRID

Die CreateMultipartUpload (früher Initiate Multipart Upload genannt) Operation initiiert einen Multipart-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 (dies wird durch ILM bestimmt).

Wenn die ILM-Regel, die auf ein aufgenommenes Objekt zutrifft, den Strict "Aufnahmeoption"-Typ verwendet, hat der `x-amz-storage-class` Header keine Auswirkung.

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

- `STANDARD` (Standard)
  - **Dual commit**: Wenn die ILM-Regel die Dual commit-Ingest-Option festlegt, wird unmittelbar nach der Aufnahme eines Objekts eine zweite Kopie dieses Objekts erstellt und auf einen anderen Storage Node verteilt (dual commit). Bei der Auswertung der ILM prüft StorageGRID, ob diese anfänglichen Zwischenkopien den Platzierungsanweisungen in der Regel entsprechen. Wenn dies nicht der Fall ist, müssen möglicherweise neue Objektkopien an anderen Standorten erstellt und die anfänglichen Zwischenkopien gelöscht werden.
  - **Ausgeglichen**: Wenn die ILM-Regel die Option „Ausgeglichen“ vorsieht und StorageGRID nicht sofort alle in der Regel angegebenen Kopien erstellen kann, erstellt StorageGRID zwei Zwischenkopien auf verschiedenen Storage Nodes.

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

- `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 Auswirkung. Die `REDUCED_REDUNDANCY` Option ist am besten geeignet, wenn die ILM-Regel, die auf das Objekt zutrifft, eine einzelne replizierte Kopie erstellt. In diesem Fall wird durch die Verwendung von `REDUCED_REDUNDANCY` diese Option das unnötige Erstellen und Löschen einer zusätzlichen Objektkopie bei jedem Ingest-Vorgang vermieden.

  Die Verwendung der `REDUCED_REDUNDANCY` Option wird unter anderen Umständen nicht empfohlen. `REDUCED_REDUNDANCY` Das Risiko von Objekt-Datenverlust während der Aufnahme wird dadurch erhöht. Beispielsweise kann es zu Datenverlust kommen, wenn die einzige Kopie zunächst auf einem Storage Node gespeichert wird, der ausfällt, bevor die ILM-Auswertung erfolgen kann.

> Wenn für einen bestimmten Zeitraum nur eine replizierte Kopie vorhanden ist, besteht das Risiko eines dauerhaften Datenverlusts. Wenn nur eine replizierte Kopie eines Objekts existiert, geht dieses Objekt verloren, wenn ein Storage Node ausfällt oder einen schwerwiegenden Fehler aufweist. Auch während Wartungsarbeiten wie Upgrades ist der Zugriff auf das Objekt vorübergehend nicht möglich.

Die Angabe von `REDUCED_REDUNDANCY` wirkt sich lediglich darauf aus, wie viele Kopien beim erstmaligen Einlesen eines Objekts erstellt werden. Sie hat keinen Einfluss darauf, wie viele Kopien des Objekts bei der Auswertung durch die aktiven ILM-Richtlinien erstellt werden und führt nicht dazu, dass Daten im StorageGRID System mit geringerer Redundanz gespeichert werden.

> Wenn Sie ein Objekt in einen Bucket mit aktiviertem S3 Object Lock importieren, wird die `REDUCED_REDUNDANCY` Option ignoriert. Wenn Sie ein Objekt in einen älteren Compliant-Bucket importieren, gibt die `REDUCED_REDUNDANCY` Option einen Fehler zurück. StorageGRID führt immer einen Dual-Commit-Import durch, um die Einhaltung der Compliance-Anforderungen zu gewährleisten.

Unterstützte Anfrageheader

Die folgenden Anfrage-Header werden unterstützt:

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

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

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

  Beim Festlegen des Namen-Wert-Paares für benutzerdefinierte Metadaten wird dieses allgemeine Format verwendet:

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

  Wenn die Option **Benutzerdefinierte Erstellungszeit** als Referenzzeit für eine ILM-Regel verwendet werden soll, muss `creation-time` als Name der Metadaten verwendet werden, die den Erstellungszeitpunkt des Objekts aufzeichnen. Beispielsweise:

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

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

  > Das Hinzufügen von `creation-time` als benutzerdefinierte Metadaten ist nicht zulässig, wenn ein Objekt zu einem Bucket hinzugefügt wird, für den die Legacy Compliance aktiviert ist. Ein Fehler wird zurückgegeben.

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

    Wird eine Anfrage ohne diese Header gestellt, werden die Standard-Aufbewahrungseinstellungen des Buckets verwendet, um das Aufbewahrungsdatum der Objektversion zu berechnen.

  "S3 REST API zur Konfiguration von S3 Object Lock verwenden"

- 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`

    Anfrage-Header für serverseitige Verschlüsselung

    > Informationen dazu, wie StorageGRID UTF-8-Zeichen verarbeitet, siehe "PutObject".

Anfrage-Header für serverseitige Verschlüsselung

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

- **SSE**: Der folgende Header ist in der CreateMultipartUpload-Anfrage zu verwenden, wenn das Objekt mit einem eindeutigen, von StorageGRID verwalteten Schlüssel verschlüsselt werden soll. Dieser Header darf in keiner der UploadPart-Anfragen angegeben werden.
  - `x-amz-server-side-encryption`
- **SSE-C**: Alle drei dieser Header sind in der CreateMultipartUpload-Anfrage (und in jeder nachfolgenden UploadPart-Anfrage) zu verwenden, wenn das Objekt mit einem eindeutigen Schlüssel verschlüsselt werden soll, 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`: Der MD5-Digest des Verschlüsselungsschlüssels des neuen Objekts ist anzugeben.

> Die von Ihnen bereitgestellten Verschlüsselungsschlüssel werden niemals gespeichert. Wenn Sie einen Verschlüsselungsschlüssel verlieren, verlieren Sie das zugehörige Objekt. Vor der Verwendung von kundenseitig bereitgestellten Schlüsseln zur Sicherung von Objektdaten sollten die Hinweise zu "Verwendung serverseitiger Verschlüsselung" beachtet werden.

Nicht unterstützte Anfrageheader

Der folgende Anfrageheader wird nicht unterstützt:

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

  Der `x-amz-website-redirect-location`Header gibt `XNotImplemented` zurück.

Versionierung

Ein Multipart-Upload besteht aus separaten Operationen zum Initiieren des Uploads, Auflisten der Uploads, Hochladen der einzelnen Teile, Zusammenfügen der hochgeladenen Teile und Abschließen des Uploads. Objekte werden erstellt (und gegebenenfalls versioniert), wenn die CompleteMultipartUpload-Operation ausgeführt wird.

S3 ListMultipartUploads Operation und unterstützte Parameter in StorageGRID

Der ListMultipartUploads-Vorgang listet laufende Multipart-Uploads für einen Bucket auf.

Die folgenden Anfrageparameter werden unterstützt:

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

- `upload-id-marker`
- `Host`
- `Date`
- `Authorization`

Versionierung

Ein Multipart-Upload besteht aus separaten Operationen zum Initiieren des Uploads, Auflisten der Uploads, Hochladen der einzelnen Teile, Zusammenfügen der hochgeladenen Teile und Abschließen des Uploads. Objekte werden erstellt (und gegebenenfalls versioniert), wenn die CompleteMultipartUpload-Operation ausgeführt wird.

Unterstützte und nicht unterstützte Anforderungsheader für S3 UploadPart-Operationen in StorageGRID

Der UploadPart-Vorgang lädt einen Teil eines Multipart-Uploads für ein Objekt hoch.

Unterstützte Anfrageheader

Die folgenden Anfrage-Header werden unterstützt:

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

Anfrage-Header für serverseitige Verschlüsselung

Wenn Sie für die CreateMultipartUpload Anfrage die SSE-C-Verschlüsselung angegeben haben, müssen Sie außerdem die folgenden Anfrageheader in jede UploadPart Anfrage 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 Anfrage angegeben haben.
- `x-amz-server-side-encryption-customer-key-MD5`: Geben Sie denselben MD5-Digest an, den Sie in der CreateMultipartUpload Anfrage angegeben haben.

> Die von Ihnen bereitgestellten Verschlüsselungsschlüssel werden niemals gespeichert. Wenn ein Verschlüsselungsschlüssel verloren geht, geht auch das zugehörige Objekt verloren. Vor der Verwendung von kundenseitig bereitgestellten Schlüsseln zur Sicherung von Objektdaten sind die Hinweise in "Serverseitige Verschlüsselung verwenden" zu beachten.

Wenn Sie während der CreateMultipartUpload-Anfrage eine SHA-256-Prüfsumme angegeben haben, muss in jeder UploadPart-Anfrage auch der folgende Anfrage-Header enthalten sein:

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

Nicht unterstützte Anfrageheader

Die folgenden Anfrageheader werden nicht unterstützt:

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

Versionierung

Ein Multipart-Upload besteht aus separaten Operationen zum Initiieren des Uploads, Auflisten der Uploads, Hochladen der einzelnen Teile, Zusammenfügen der hochgeladenen Teile und Abschließen des Uploads. Objekte werden erstellt (und gegebenenfalls versioniert), wenn die CompleteMultipartUpload-Operation ausgeführt wird.

Laden Sie einen Teil eines Objekts in StorageGRID mit der S3 UploadPartCopy-Operation hoch

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

Die UploadPartCopy-Operation ist mit dem gesamten Verhalten der Amazon S3 REST API implementiert. Änderungen vorbehalten.

Diese Anfrage liest und schreibt die im `x-amz-copy-source-range`StorageGRID System angegebenen Objektdaten.

Die folgenden Anfrage-Header 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`

Anfrage-Header für serverseitige Verschlüsselung

Wenn Sie für die CreateMultipartUpload Anfrage die SSE-C-Verschlüsselung angegeben haben, müssen die folgenden Anfrage-Header auch in jede UploadPartCopy Anfrage aufgenommen werden:

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

Wenn das Quellobjekt mit einem vom Kunden bereitgestellten Schlüssel (SSE-C) verschlüsselt ist, sind die folgenden drei Header in die UploadPartCopy-Anfrage aufzunehmen, damit das Objekt entschlüsselt und anschließend kopiert werden kann:

- `x-amz-copy-source-server-side-encryption-customer-algorithm`: Geben Sie `AES256` an.
- `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 ein Verschlüsselungsschlüssel verloren geht, geht auch das zugehörige Objekt verloren. Vor der Verwendung von kundenseitig bereitgestellten Schlüsseln zur Sicherung von Objektdaten sind die Hinweise in "Serverseitige Verschlüsselung verwenden" zu beachten.

Versionierung

Ein Multipart-Upload besteht aus separaten Operationen zum Initiieren des Uploads, Auflisten der Uploads, Hochladen der einzelnen Teile, Zusammenfügen der hochgeladenen Teile und Abschließen des Uploads. Objekte werden erstellt (und gegebenenfalls versioniert), wenn die CompleteMultipartUpload-Operation ausgeführt wird.

StorageGRID S3 REST-API-Fehlerantworten

Das StorageGRID System unterstützt alle Standard-Fehlerantworten der S3 REST API, die zutreffen. Zusätzlich ergänzt die StorageGRID Implementierung mehrere benutzerdefinierte Antworten.

Unterstützte S3 API-Fehlercodes

Name	HTTP Status
AccessDenied	403 Forbidden
BadDigest	400 Fehlerhafte Anfrage
BucketAlreadyExists	409 Konflikt
BucketNotEmpty	409 Konflikt
IncompleteBody	400 Fehlerhafte Anfrage
InternalError	500 Interner Serverfehler
InvalidAccessKeyId	403 Forbidden
InvalidArgument	400 Fehlerhafte Anfrage
InvalidBucketName	400 Fehlerhafte Anfrage
InvalidBucketState	409 Konflikt
InvalidDigest	400 Fehlerhafte Anfrage
InvalidEncryptionAlgorithmError	400 Fehlerhafte Anfrage
InvalidPart	400 Fehlerhafte Anfrage

Name	HTTP Status
InvalidPartOrder	400 Fehlerhafte Anfrage
InvalidRange	416 Angefragter Bereich nicht erfüllbar
InvalidRequest	400 Fehlerhafte Anfrage
InvalidStorageClass	400 Fehlerhafte Anfrage
InvalidTag	400 Fehlerhafte Anfrage
Ungültige URI	400 Fehlerhafte Anfrage
KeyTooLong	400 Fehlerhafte Anfrage
Fehlerhaftes XML	400 Fehlerhafte Anfrage
MetadataTooLarge	400 Fehlerhafte Anfrage
MethodNotAllowed	405 Methode nicht erlaubt
MissingContentLength	411 Länge erforderlich
MissingRequestBodyError	400 Fehlerhafte Anfrage
MissingSecurityHeader	400 Fehlerhafte Anfrage
NoSuchBucket	404 Nicht gefunden
NoSuchKey	404 Nicht gefunden
NoSuchUpload	404 Nicht gefunden
NotImplemented	501 Nicht implementiert
NoSuchBucketPolicy	404 Nicht gefunden
ObjectLockConfigurationNotFoundError	404 Nicht gefunden
PreconditionFailed	412 Vorbedingung nicht erfüllt
RequestTimeTooSkewed	403 Forbidden
ServiceUnavailable	503 Dienst nicht verfügbar

Name	HTTP Status
SignatureDoesNotMatch	403 Forbidden
TooManyBuckets	400 Fehlerhafte Anfrage
UserKeyMustBeSpecified	400 Fehlerhafte Anfrage

StorageGRID benutzerdefinierte Fehlercodes

Name	Beschreibung	HTTP Status
XBucketLifecycleNotAllowed	Die Bucket-Lifecycle-Konfiguration ist in einem Legacy Compliant Bucket nicht zulässig.	400 Fehlerhafte Anfrage
XBucketPolicyParseException	Fehler beim Parsen des empfangenen Bucket-Policy-JSON.	400 Fehlerhafte Anfrage
XComplianceConflict	Vorgang aufgrund veralteter Compliance-Einstellungen abgelehnt.	403 Forbidden
XComplianceReducedRedundancyForbidden	Reduzierte Redundanz ist in älteren Compliant Buckets nicht zulässig.	400 Fehlerhafte Anfrage
XMaxBucketPolicyLengthExceeded	Ihre Richtlinie überschreitet die maximal zulässige Bucket-Richtlinienlänge.	400 Fehlerhafte Anfrage
XMissingInternalRequestHeader	Es fehlt ein Header einer internen Anfrage.	400 Fehlerhafte Anfrage
XNoSuchBucketCompliance	Für den angegebenen Bucket ist die Legacy Compliance nicht aktiviert.	404 Nicht gefunden
XNotAcceptable	Die Anfrage enthält einen oder mehrere Accept-Header, die nicht erfüllt werden konnten.	406 Nicht akzeptabel
XNotImplemented	Die von Ihnen bereitgestellte Anfrage impliziert eine Funktionalität, die nicht implementiert ist.	501 Nicht implementiert

StorageGRID benutzerdefinierte Operationen

Unterstützte benutzerdefinierte Bucket-Operationen in StorageGRID

Das StorageGRID System unterstützt benutzerdefinierte Operationen, die zur S3 REST

API hinzugefügt werden.

Die folgende Tabelle listet die von StorageGRID unterstützten benutzerdefinierten Vorgänge auf.

Betrieb	Beschreibung
"GET Bucket Konsistenz"	Gibt die Konsistenz zurück, die auf einen bestimmten Bucket angewendet wird.
"PUT Bucket Konsistenz"	Legt die Konsistenz fest, die auf einen bestimmten Bucket angewendet wird.
"Letzte Zugriffszeit des Buckets abrufen"	Gibt zurück, ob die Aktualisierung der letzten Zugriffszeit für einen bestimmten Bucket aktiviert oder deaktiviert ist.
"PUT Bucket letzter Zugriffszeitpunkt"	Ermöglicht das Aktivieren oder Deaktivieren von Aktualisierungen der letzten Zugriffszeit für einen bestimmten Bucket.
"Bucket-Metadatenbenachrichtigungskonfiguration löschen"	Löscht die XML-Konfiguration für Metadatenbenachrichtigungen, die einem bestimmten Bucket zugeordnet ist.
"GET Bucket-Metadaten-Benachrichtigungskonfiguration"	Gibt die XML-Konfiguration für Metadatenbenachrichtigungen zurück, die einem bestimmten Bucket zugeordnet ist.
"PUT Bucket Metadaten-Benachrichtigungskonfiguration"	Konfiguriert den Metadaten-Benachrichtigungsdienst für einen Bucket.
"Speichernutzung abrufen"	Zeigt die gesamte Speichernutzung eines Kontos sowie für jeden mit dem Konto verknüpften Bucket an.
"Veraltet: CreateBucket mit Konformitätseinstellungen"	Veraltet und nicht mehr unterstützt: Es können keine neuen Buckets mehr mit aktivierter Compliance erstellt werden.
"Veraltet: GET Bucket Konformität"	Veraltet, aber weiterhin unterstützt: Gibt die aktuell für einen bestehenden Legacy Compliant Bucket geltenden Compliance-Einstellungen zurück.
"Veraltet: PUT Bucket Compliance"	Veraltet, aber weiterhin unterstützt: Ermöglicht die Änderung der Compliance-Einstellungen für einen bestehenden Legacy Compliant Bucket.

Den Bucket-Konsistenzgrad in StorageGRID mit der GET-Anfrage für Bucket-Konsistenz abrufen

Die GET Bucket consistency-Anfrage ermöglicht die Bestimmung der Konsistenz, die auf einen bestimmten Bucket angewendet wird.

Die Standardkonsistenz ist so eingestellt, dass für neu erstellte Objekte Lese-nach-Schreib-Vorgänge

garantiert werden.

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

Anfragebeispiel

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

Antwort

In der Antwort-XML, `<Consistency>` wird einer der folgenden Werte zurückgegeben:

Konsistenz	Beschreibung
alle	Alle Knoten empfangen die Daten sofort, oder die Anfrage schlägt fehl.
stark global	Gewährleistet Lese-nach-Schreib-Konsistenz für alle Clientanfragen an allen Standorten.
starker Standort	Garantiert Lese-nach-Schreib-Konsistenz für alle Client-Anfragen innerhalb eines Standorts.
Lesen nach neuem Schreiben	(Standardeinstellung) Bietet Lese-nach-Schreib-Konsistenz für neue Objekte und letztendliche Konsistenz für Objektaktualisierungen. Hohe Verfügbarkeit und Garantien für den Datenschutz werden geboten. Für die meisten Fälle empfohlen.
verfügbar	Gewährleistet letztendliche Konsistenz sowohl für neue Objekte als auch für Objektaktualisierungen. Für S3-Buckets sollte dies nur bei Bedarf verwendet werden (zum Beispiel für einen Bucket, der Log-Werte enthält, die selten gelesen werden, oder für HEAD- oder GET-Operationen auf Schlüsseln, die nicht existieren). Nicht unterstützt für S3 FabricPool Buckets.

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>
```

Verwandte Informationen

"Konsistenz"

Geben Sie Konsistenzstufen in StorageGRID mit der PUT Bucket consistency-Anforderung an

Die PUT Bucket Konsistenzanforderung ermöglicht die Angabe der Konsistenz, die auf Operationen angewendet wird, die an einem Bucket durchgeführt werden.

Die Standardkonsistenz ist so eingestellt, dass für neu erstellte Objekte Lese-nach-Schreib-Vorgänge garantiert werden.

Bevor Sie beginnen

Sie müssen über die Berechtigung s3:PutBucketConsistency verfügen oder als Root-Benutzer angemeldet sein, um diesen Vorgang abzuschließen.

Anfrage

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

Konsistenz	Beschreibung
alle	Alle Knoten empfangen die Daten sofort, oder die Anfrage schlägt fehl.
stark global	Gewährleistet Lese-nach-Schreib-Konsistenz für alle Clientanfragen an allen Standorten.
starker Standort	Garantiert Lese-nach-Schreib-Konsistenz für alle Client-Anfragen innerhalb eines Standorts.
Lesen nach neuem Schreiben	(Standardeinstellung) Bietet Lese-nach-Schreib-Konsistenz für neue Objekte und letztendliche Konsistenz für Objektaktualisierungen. Hohe Verfügbarkeit und Garantien für den Datenschutz werden geboten. Für die meisten Fälle empfohlen.

Konsistenz	Beschreibung
verfügbar	Gewährleistet letztendliche Konsistenz sowohl für neue Objekte als auch für Objektaktualisierungen. Für S3-Buckets sollte dies nur bei Bedarf verwendet werden (zum Beispiel für einen Bucket, der Log-Werte enthält, die selten gelesen werden, oder für HEAD- oder GET-Operationen auf Schlüsseln, die nicht existieren). Nicht unterstützt für S3 FabricPool Buckets.

Hinweis: Im Allgemeinen ist die Konsistenz „Read-after-new-write“ zu verwenden. Wenn Anfragen nicht korrekt funktionieren, sollte nach Möglichkeit das Verhalten des Anwendungsclients geändert werden. Alternativ kann der Client so konfiguriert werden, dass er die Konsistenz für jede API-Anfrage angibt. Die Konsistenz auf Bucket-Ebene sollte nur als letzte Möglichkeit festgelegt werden.

Anfragebeispiel

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

Verwandte Informationen

"Konsistenz"

Rufen Sie den Status der letzten Zugriffszeit eines Buckets in StorageGRID mit der Anfrage „GET Bucket last access time“ ab.

Die GET Bucket last access time-Anfrage ermöglicht die Feststellung, ob Aktualisierungen der letzten Zugriffszeit für einzelne Buckets aktiviert oder deaktiviert sind.

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

Anfragebeispiel

```
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>
```

Aktivieren und Deaktivieren von Aktualisierungen der letzten Zugriffszeit in StorageGRID mit der PUT Bucket last access time-Anfrage

Die PUT Bucket Last Access Time-Anfrage ermöglicht das Aktivieren oder Deaktivieren der Aktualisierung der letzten Zugriffszeit für einzelne Buckets. Das Deaktivieren der Aktualisierung der letzten Zugriffszeit verbessert die Performance 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 Kontoinhaber (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 besitzen, die mit einer früheren Version von StorageGRID erstellt wurden und das neue Standardverhalten übernehmen möchten, müssen Sie die Aktualisierungen der letzten Zugriffszeit für jeden dieser älteren Buckets explizit deaktivieren. Aktualisierungen der letzten Zugriffszeit können mit der PUT Bucket last access time-Anfrage oder auf der Detailseite eines Buckets im Tenant Manager aktiviert oder deaktiviert werden. Siehe "Aktualisierungen der letzten Zugriffszeit aktivieren oder deaktivieren".

Wenn die Aktualisierung der letzten Zugriffszeit für einen Bucket deaktiviert ist, gilt bei Operationen am Bucket das folgende Verhalten:

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

Deaktivieren Sie den Suchintegrationsdienst mit der DELETE Bucket-Metadaten-Benachrichtigungskonfigurationsanfrage in StorageGRID

Die DELETE Bucket Metadatenbenachrichtigungskonfigurationsanfrage ermöglicht das Deaktivieren des Suchintegrationsdienstes für einzelne Buckets durch Löschen der Konfigurations-XML.

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

Anfragebeispiel

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
```

Konfigurieren Sie die Suchintegration in StorageGRID mit der GET Bucket-Metadatenbenachrichtigungskonfigurationsanfrage

Mit der GET-Anfrage zur Konfiguration der Bucket-Metadatenbenachrichtigung kann die Konfigurations-XML abgerufen werden, die zur Konfiguration der Suchintegration für einzelne Buckets verwendet wird.

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

Anfragebeispiel

Diese Anfrage ruft die Metadatenbenachrichtigungskonfiguration für den Bucket mit dem Namen `bucket` ab.

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

Antwort

Der Antworttext enthält die Metadaten-Benachrichtigungskonfiguration für den Bucket. Die Metadaten-Benachrichtigungskonfiguration ermöglicht es, festzulegen, wie der Bucket für die Suchintegration konfiguriert ist. Das bedeutet, es kann bestimmt werden, welche Objekte indiziert werden und an welche Endpunkte deren 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 enthält eine oder mehrere Regeln. Jede Regel legt die Objekte fest, auf die sie angewendet wird, sowie das Ziel, an das StorageGRID die Objektmetadaten senden soll. Ziele müssen mithilfe der URN eines StorageGRID Endpunkts angegeben werden.

Name	Beschreibung	Erforderlich
MetadataNotificationConfiguration	Container-Tag für Regeln, die zur Angabe der Objekte und des Ziels für Metadatenbenachrichtigungen verwendet werden. Enthält ein oder mehrere Regelelemente.	Ja

Name	Beschreibung	Erforderlich
Regel	Container-Tag für eine Regel, die die Objekte identifiziert, deren Metadaten zu einem bestimmten Index hinzugefügt werden sollen. Regeln mit sich überschneidenden Präfixen werden abgelehnt. Im MetadataNotificationConfiguration-Element enthalten.	Ja
ID	Eindeutige Kennung für die Regel. Im Regelelement enthalten.	Nein
Status	Der Status kann „Aktiviert“ oder „Deaktiviert“ sein. Für deaktivierte Regeln wird keine Aktion durchgeführt. 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, ein leeres Präfix angeben. Im Regelelement enthalten.	Ja
Ziel	Container-Tag für das Ziel einer Regel. Im Regelelement enthalten.	Ja

Name	Beschreibung	Erforderlich
Urne	URN des Ziels, an das 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 dem Typ enden, unter 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 folgendes Format: • `arn:aws:es:_region:account-ID_:domain/mydomain/myindex/mytype` • `urn:mysite:es:::mydomain/myindex/mytype` Der Endpunkt muss konfiguriert sein, bevor die Konfigurations-XML übermittelt wird, sonst schlägt die Konfiguration mit einem 404-Fehler fehl. Urn ist im Element „Ziel“ enthalten.	Ja

Antwortbeispiel

Der zwischen den `<MetadataNotificationConfiguration></MetadataNotificationConfiguration>` Tags enthaltene XML-Code zeigt, wie die Integration mit einem Suchintegrationsendpunkt für den Bucket konfiguriert ist. In diesem Beispiel werden Objektmetadaten an einen Elasticsearch Index mit dem Namen `current` und dem Typ mit dem Namen `2017` gesendet, der in einer AWS Domäne mit dem Namen `records` gehostet wird.

```
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>
```

Verwandte Informationen

"Ein Mandantenkonto verwenden"_

Aktivieren Sie den Suchintegrationsdienst in StorageGRID mit der PUT Bucket-Metadatenbenachrichtigungs-Konfigurationsanfrage

Die PUT-Anfrage zur Konfiguration der Bucket-Metadatenbenachrichtigung ermöglicht die Aktivierung des Suchintegrationsdienstes für einzelne Buckets. Die im Anfragetext angegebene XML-Konfiguration für die Metadatenbenachrichtigung legt die Objekte fest, deren Metadaten an den Zielsuchindex gesendet werden.

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

Anfrage

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

Objekte können anhand des Präfixes ihres Objektnamens gefiltert werden. Beispielsweise könnten Metadaten für Objekte mit dem Präfix `/images`an ein Ziel und Objekte mit dem Präfix `/videos`an ein anderes Ziel gesendet werden.

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

Ziele müssen mithilfe der URN eines StorageGRID Endpunkts angegeben werden. Der Endpunkt muss zum Zeitpunkt der Übermittlung der Metadatenbenachrichtigungskonfiguration vorhanden sein, andernfalls schlägt die Anfrage fehl als `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 für Metadatenbenachrichtigungen.

Name	Beschreibung	Erforderlich
MetadataNotificationConfiguration	Container-Tag für Regeln, die zur Angabe 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 zu einem bestimmten Index hinzugefügt werden sollen. Regeln mit sich überschneidenden Präfixen werden abgelehnt. Im MetadataNotificationConfiguration-Element enthalten.	Ja
ID	Eindeutige Kennung für die Regel. Im Regelelement enthalten.	Nein

Name	Beschreibung	Erforderlich
Status	Der Status kann „Aktiviert“ oder „Deaktiviert“ sein. Für deaktivierte Regeln wird keine Aktion durchgeführt. 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, ein leeres Präfix angeben. Im Regelelement enthalten.	Ja
Ziel	Container-Tag für das Ziel einer Regel. Im Regelelement enthalten.	Ja
Urne	URN des Ziels, an das 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 dem Typ enden, unter 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 folgendes Format: • `arn:aws:es:region:account-ID:domain/mydomain/myindex/mytype` • `urn:mysite:es:::mydomain/myindex/mytype` Der Endpunkt muss konfiguriert sein, bevor die Konfigurations-XML übermittelt wird, sonst schlägt die Konfiguration mit einem 404-Fehler fehl. Urn ist im Element „Ziel“ enthalten.	Ja

Anforderungsbeispiele

Dieses Beispiel zeigt die Aktivierung der Suchintegration für einen Bucket. In diesem Beispiel werden die Objektmetadaten aller 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, die dem Präfix `/images` entsprechen, an ein Ziel gesendet, während Objektmetadaten für Objekte, die dem Präfix `/videos` entsprechen, an ein zweites Ziel gesendet werden.

```
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 der Suchintegrationsdienst für einen Bucket aktiviert ist, 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 für das JSON, das generiert werden könnte, wenn ein Objekt mit dem Schlüssel `SGWS/Tagging.txt` in einem Bucket namens `test` erstellt wird. Der `test` Bucket ist nicht versioniert, daher ist das `versionId` Tag 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

Die Tabelle listet alle Felder auf, die im JSON-Dokument enthalten sind, das an den Zielendpunkt gesendet wird, wenn die Suchintegration aktiviert ist.

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

Typ	Artikelname	Beschreibung
Bucket und Objektinformationen	Bucket	Name des Buckets
Bucket und Objektinformationen	Schlüssel	Objektschlüsselname
Bucket und Objektinformationen	versionID	Objektversion, für Objekte in versionierten Buckets
Bucket und Objektinformationen	Region	Bucket Region, zum Beispiel `us-east-1`
Systemmetadaten	Größe	Objektgröße (in Bytes) aus Sicht eines HTTP-Clients
Systemmetadaten	md5	Objekt-Hashwert
Benutzermetadaten	Metadaten *`key:value`*	Alle Benutzermetadaten für das Objekt als Schlüssel-Wert-Paare
Tags	Tags *`key:value`*	Alle für das Objekt definierten Objekt-Tags als Schlüssel-Wert-Paare

Für Tags und Benutzermetadaten übergibt StorageGRID Datums- und Zahlenwerte als Zeichenketten oder als S3-Ereignisbenachrichtigungen an Elasticsearch. Damit Elasticsearch diese Zeichenketten als Datums- oder Zahlenwerte interpretiert, sind die Anweisungen von Elasticsearch zur dynamischen Feldzuordnung und zur Zuordnung von Datumsformaten zu beachten. Die dynamischen Feldzuordnungen müssen im Index aktiviert sein, bevor der Suchintegrationsdienst konfiguriert wird. Nachdem ein Dokument indiziert wurde, können die Feldtypen des Dokuments im Index nicht mehr bearbeitet werden.

Verwandte Informationen

"Ein Mandantenkonto verwenden"_

Abrufen der Speichernutzung von Konto und Bucket in StorageGRID mit der GET Storage Usage-Anfrage

Die GET Storage Usage-Anfrage gibt die Gesamtmenge des von einem Konto verwendeten Speicherplatzes sowie für jeden mit dem Konto verknüpften Bucket an.

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

Standardmäßig versucht StorageGRID, Nutzungsinformationen mittels starker globaler Konsistenz abzurufen. Kann keine starke globale Konsistenz erreicht werden, versucht StorageGRID, die Nutzungsinformationen mit starker Standortkonsistenz abzurufen.

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

Anfragebeispiel

```
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 Byte 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 zu den `ObjectCount` und `DataBytes` Werten in der Antwort bei. Löschmarkierungen werden nicht zur `ObjectCount` Gesamtzahl hinzugezählt.

Verwandte Informationen

"Konsistenz"

Veraltete Bucket-Anfragen für Legacy Compliance

Veraltete Bucket-Anfragen für die Legacy StorageGRID Compliance

Möglicherweise ist die Verwendung der StorageGRID S3 REST API erforderlich, um Buckets zu verwalten, die mit der älteren Compliance-Funktion erstellt wurden.

Compliance-Funktion veraltet

Die in früheren StorageGRID Versionen verfügbare StorageGRID Compliance Funktion ist veraltet und wurde durch S3 Object Lock ersetzt.

Wenn Sie zuvor die globale Compliance-Einstellung aktiviert hatten, ist die globale S3 Object Lock-Einstellung in StorageGRID 11.6 aktiviert. Es können keine neuen Buckets mehr mit aktivierter Compliance erstellt werden; jedoch kann bei Bedarf die StorageGRID S3 REST API verwendet werden, um vorhandene Legacy Compliant Buckets zu verwalten.

- "S3 REST API zur Konfiguration von S3 Object Lock verwenden"
- "Objekte mit ILM verwalten"
- "NetApp Knowledge Base: Verwaltung älterer konformer Buckets in StorageGRID 11.5"

Veraltete Compliance-Anfragen:

- "Veraltet – Änderungen an PUT Bucket-Anfragen zur Einhaltung der Vorschriften"

  Das XML-Element SGCompliance ist veraltet. Früher konnte dieses benutzerdefinierte StorageGRID Element im optionalen XML-Anforderungstext von PUT Bucket-Anfragen eingefügt werden, um einen konformen Bucket zu erstellen.

- "Veraltet – GET Bucket Konformität"

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

- "Veraltet – PUT Bucket Konformität"

  Die PUT Bucket compliance-Anfrage ist veraltet. Sie können diese Anfrage jedoch weiterhin verwenden, um die Compliance-Einstellungen eines bestehenden Legacy Compliant-Buckets zu ändern. Beispielsweise kann ein bestehender Bucket unter rechtliche Aufbewahrung gestellt oder seine Aufbewahrungsfrist verlängert werden.

Veraltetes SGCompliance-Element in StorageGRID

Das XML-Element SGCompliance ist veraltet. Früher konnte dieses benutzerdefinierte StorageGRID-Element im optionalen XML-Anforderungstext von CreateBucket-Anfragen zur Erstellung eines konformen Buckets einbezogen werden.

> Die in früheren StorageGRID Versionen verfügbare StorageGRID Compliance-Funktion ist veraltet und wurde durch S3 Object Lock ersetzt. Weitere Details finden Sie hier:
>
> - "S3 REST API zur Konfiguration von S3 Object Lock verwenden"
> - "NetApp Knowledge Base: Verwaltung älterer konformer Buckets in StorageGRID 11.5"

Es können keine neuen Buckets mehr mit aktivierter Compliance erstellt werden. Die folgende Fehlermeldung wird zurückgegeben, wenn versucht wird, mithilfe der CreateBucket-Anforderungsänderungen für Compliance einen neuen Compliant Bucket zu erstellen:

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

Veraltete GET Bucket Compliance-Anfrage in StorageGRID

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

> Die in früheren StorageGRID Versionen verfügbare StorageGRID Compliance-Funktion ist veraltet und wurde durch S3 Object Lock ersetzt. Weitere Details finden Sie hier:
>
> - "S3 REST API zur Konfiguration von S3 Object Lock verwenden"
> - "NetApp Knowledge Base: Verwaltung älterer konformer Buckets in StorageGRID 11.5"

Sie müssen über die Berechtigung s3:GetBucketCompliance verfügen oder als Account Root angemeldet sein, um diesen Vorgang abzuschließen.

Anfragebeispiel

Dieses Beispiel ermöglicht die Bestimmung der Compliance-Einstellungen für den Bucket mit dem Namen `mybucket`.

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

Antwortbeispiel

In der Antwort-XML `<SGCompliance>` werden die für den Bucket geltenden Compliance-Einstellungen aufgeführt. Dieses Beispiel zeigt die Compliance-Einstellungen für einen Bucket, in dem jedes Objekt ab dem Zeitpunkt der Aufnahme in das Grid ein Jahr lang (525,600 Minuten) aufbewahrt wird. Für diesen Bucket besteht derzeit keine rechtliche Aufbewahrungspflicht. 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
RetentionPeriodMinutes	Die Aufbewahrungsdauer für Objekte, die diesem Bucket hinzugefügt werden, in Minuten. Die Aufbewahrungsdauer beginnt, wenn das Objekt in das Grid aufgenommen wird.
LegalHold	• Richtig: Dieser Bucket unterliegt derzeit einer rechtlichen Sperre. Objekte in diesem Bucket können nicht gelöscht werden, bis die rechtliche Sperre aufgehoben ist, selbst wenn ihre Aufbewahrungsfrist abgelaufen ist. • Falsch: Dieser Bucket unterliegt derzeit keiner rechtlichen Aufbewahrungspflicht. Objekte in diesem Bucket können gelöscht werden, wenn ihre Aufbewahrungsfrist abgelaufen ist.
AutoDelete	• Richtig: Die Objekte in diesem Bucket werden automatisch gelöscht, wenn ihre Aufbewahrungsfrist abläuft, es sei denn, der Bucket befindet sich unter einer rechtlichen Sperre. • Falsch: Die Objekte in diesem Bucket werden nicht automatisch gelöscht, wenn die Aufbewahrungsfrist abläuft. Diese Objekte müssen manuell gelöscht werden, wenn sie gelöscht werden sollen.

Fehlermeldungen

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

Veraltete PUT Bucket Compliance-Anfrage in StorageGRID

Die PUT Bucket compliance-Anfrage ist veraltet. Sie können diese Anfrage jedoch weiterhin verwenden, um die Compliance-Einstellungen eines bestehenden Legacy Compliant-Buckets zu ändern. Beispielsweise kann ein bestehender Bucket unter rechtliche Aufbewahrung gestellt oder seine Aufbewahrungsfrist verlängert werden.

> Die in früheren StorageGRID Versionen verfügbare StorageGRID Compliance-Funktion ist veraltet und wurde durch S3 Object Lock ersetzt. Weitere Details finden Sie hier:
>
> - "S3 REST API zur Konfiguration von S3 Object Lock verwenden"
> - "NetApp Knowledge Base: Verwaltung älterer konformer Buckets in StorageGRID 11.5"

Sie müssen über die Berechtigung s3:PutBucketCompliance verfügen oder als Account Root angemeldet sein, um diesen Vorgang abzuschließen.

Für jedes Feld der Compliance-Einstellungen muss beim Ausstellen einer PUT Bucket Compliance-Anfrage ein Wert angegeben werden.

Anfragebeispiel

Dieses Beispiel ändert die Compliance-Einstellungen für den Bucket mit dem Namen `mybucket`. In diesem Beispiel werden Objekte in `mybucket` nun zwei Jahre (1.051.200 Minuten) statt eines Jahres aufbewahrt, beginnend mit dem Zeitpunkt, zu dem das Objekt in das Grid aufgenommen wird. Für diesen Bucket besteht

keine rechtliche Aufbewahrungspflicht. 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
RetentionPeriodMinutes	Die Aufbewahrungsdauer für Objekte, die diesem Bucket hinzugefügt werden, in Minuten. Die Aufbewahrungsdauer beginnt, wenn das Objekt in das Grid aufgenommen wird. **Wichtig** Wenn Sie einen neuen Wert für RetentionPeriodMinutes angeben, muss dieser Wert mindestens der aktuellen Aufbewahrungsdauer des Buckets entsprechen. Nachdem die Aufbewahrungsdauer des Buckets festgelegt wurde, kann dieser Wert nicht verringert, sondern nur erhöht werden.
LegalHold	• Richtig: Dieser Bucket unterliegt derzeit einer rechtlichen Sperre. Objekte in diesem Bucket können nicht gelöscht werden, bis die rechtliche Sperre aufgehoben ist, selbst wenn ihre Aufbewahrungsfrist abgelaufen ist. • Falsch: Dieser Bucket unterliegt derzeit keiner rechtlichen Aufbewahrungspflicht. Objekte in diesem Bucket können gelöscht werden, wenn ihre Aufbewahrungsfrist abgelaufen ist.
AutoDelete	• Richtig: Die Objekte in diesem Bucket werden automatisch gelöscht, wenn ihre Aufbewahrungsfrist abläuft, es sei denn, der Bucket befindet sich unter einer rechtlichen Sperre. • Falsch: Die Objekte in diesem Bucket werden nicht automatisch gelöscht, wenn die Aufbewahrungsfrist abläuft. Diese Objekte müssen manuell gelöscht werden, wenn sie gelöscht werden sollen.

Konsistenz für Compliance-Einstellungen

Wenn die Compliance-Einstellungen für einen S3-Bucket mit einer PUT-Bucket-Compliance-Anfrage aktualisiert werden, versucht StorageGRID, die Metadaten des Buckets im gesamten Grid zu aktualisieren. Standardmäßig verwendet StorageGRID die **Strong-global** Konsistenz, um sicherzustellen, dass alle Rechenzentrumsstandorte und alle Storage Nodes, die Bucket-Metadaten enthalten, nach dem Schreiben Konsistenz für die geänderten Compliance-Einstellungen aufweisen.

Kann StorageGRID die **Strong-global** Konsistenz nicht erreichen, weil ein Rechenzentrumsstandort oder mehrere Storage Nodes an einem Standort nicht verfügbar sind, lautet der HTTP-Statuscode der Antwort `503 Service Unavailable`.

Wenn Sie diese Antwort erhalten, sollten Sie sich an den Grid-Administrator wenden, damit die benötigten Speicherdienste so schnell wie möglich bereitgestellt werden. Falls der Grid-Administrator nicht genügend Storage Nodes an jedem Standort bereitstellen kann, kann der technische Support dazu raten, die fehlgeschlagene Anfrage durch Erzwingen der **Strong-site** Konsistenz zu wiederholen.

> Die **Strong-site** Konsistenz für die PUT-Bucket-Konformität sollte nur dann erzwungen werden, wenn dies vom technischen Support ausdrücklich empfohlen wurde und die potenziellen Konsequenzen der Nutzung dieses Levels bekannt sind.

Wenn die Konsistenz auf **Strong-site** reduziert wird, garantiert StorageGRID, dass aktualisierte Compliance-Einstellungen nur für Clientanfragen innerhalb eines Standorts Lese-nach-Schreib-Konsistenz aufweisen. Dies bedeutet, dass das StorageGRID System vorübergehend mehrere, inkonsistente Einstellungen für diesen Bucket haben kann, bis alle Standorte und Storage Nodes verfügbar sind. Die inkonsistenten Einstellungen können zu unerwartetem und unerwünschtem Verhalten führen. Wenn beispielsweise ein Bucket unter eine rechtliche Aufbewahrungspflicht gestellt wird und eine niedrigere Konsistenz erzwungen wird, können die vorherigen Compliance-Einstellungen des Buckets (das heißt, rechtliche Aufbewahrungspflicht aus) an einigen Rechenzentrumsstandorten weiterhin wirksam sein. Infolgedessen können Objekte, die Ihrer Meinung nach unter rechtlicher Aufbewahrungspflicht stehen, nach Ablauf ihrer Aufbewahrungsfrist gelöscht werden, entweder durch den Benutzer oder durch AutoDelete, falls aktiviert.

Um die Verwendung der **Strong-site** Konsistenz zu erzwingen, die PUT Bucket Compliance-Anfrage erneut senden und den `Consistency-Control` HTTP-Anfrageheader wie folgt einfügen:

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

Fehlermeldungen

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

Verwandte Informationen

"Veraltet: Änderungen an PUT Bucket-Anfragen zur Einhaltung der Vorschriften"

Zugriffsrichtlinien verwalten

Wie StorageGRID AWS-Richtlinien verwendet, um den Zugriff auf S3-Buckets und Objekte zu steuern

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

Überblick über die Zugriffsrichtlinie

StorageGRID unterstützt drei Arten von Zugriffsrichtlinien:

- **Bucket-Richtlinien**, die mithilfe der GetBucketPolicy, PutBucketPolicy und DeleteBucketPolicy S3-API-Operationen oder über den Tenant Manager bzw. die Tenant Management API verwaltet werden. Bucket-Richtlinien sind Buckets zugeordnet, sodass sie den Zugriff von Benutzern im Bucket-Inhaberkonto oder in 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, sodass sie so konfiguriert werden, dass diese Gruppe auf bestimmte Ressourcen zugreifen kann, die diesem Konto gehören. Eine Gruppenrichtlinie gilt nur für eine Gruppe und möglicherweise für mehrere Buckets.
- **Sitzungsrichtlinien**, die im Rahmen einer AssumeRole-Anfrage übermittelt werden. Sitzungsrichtlinien gelten nur für die jeweilige Sitzung und definieren zusätzlich zu den durch Gruppen- und Bucket-Richtlinie gewährten Berechtigungen die Berechtigungen des Benutzers.

> Es besteht kein Unterschied in der Priorität zwischen Gruppenrichtlinien, Bucket-Richtlinien und Sitzungsrichtlinien.

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

- Statement-ID (Sid) (optional)
- Wirkung
- Principal/NotPrincipal
- Ressource/NotResource
- Aktion/NotAction
- Zustand (optional)

Richtlinienanweisungen werden mithilfe dieser Struktur erstellt, um Berechtigungen festzulegen: Grant <Effect> zum Erlauben/Verweigern von <Principal> für die Ausführung von <Action> auf <Resource>, wenn <Condition> zutrifft.

Jedes Richtlinienelement dient einer bestimmten Funktion:

Element	Beschreibung
Sid	Das Sid-Element ist optional. Das Sid dient lediglich als Beschreibung für den Benutzer. Es wird gespeichert, aber vom StorageGRID System nicht interpretiert.
Wirkung	Das Effect-Element dient dazu festzustellen, ob die angegebenen Operationen erlaubt oder verweigert werden. Operationen, die auf Buckets oder Objekten erlaubt (oder verweigert) werden, sind mit den unterstützten Schlüsselwörtern des Action-Elements zu kennzeichnen.

Element	Beschreibung
Principal/NotPrincipal	Sie können Benutzern, Gruppen und Konten den Zugriff auf bestimmte Ressourcen und die Durchführung bestimmter Aktionen ermöglichen. Wenn keine S3-Signatur in der Anfrage enthalten ist, wird anonymer Zugriff durch Angabe des Platzhalterzeichens (*) als Principal erlaubt. Standardmäßig hat nur das Konto-Root Zugriff auf Ressourcen, die dem Konto gehören. Sie müssen lediglich das Principal-Element in einer Bucket-Richtlinie angeben. Bei Gruppenrichtlinien ist die Gruppe, der die Richtlinie zugeordnet ist, das implizite Principal-Element.
Ressource/NotResource	Das Resource-Element identifiziert Buckets und Objekte. Berechtigungen für Buckets und Objekte können mithilfe des Amazon Resource Name (ARN) zur Identifizierung der Ressource erlaubt oder verweigert werden.
Aktion/NotAction	Die Elemente „Aktion“ und „Wirkung“ bilden die beiden Komponenten von Berechtigungen. Wenn eine Gruppe eine Ressource anfordert, wird ihr der Zugriff entweder gewährt oder verweigert. Der Zugriff wird verweigert, sofern Sie keine Berechtigungen explizit zuweisen, jedoch kann eine durch eine andere Richtlinie gewährte Berechtigung durch eine explizite Verweigerung überschrieben werden.
Zustand	Das Element „Bedingung“ ist optional. Bedingungen ermöglichen das Erstellen von Ausdrücken, um festzulegen, wann eine Richtlinie angewendet werden soll.

Im Action-Element kann das Platzhalterzeichen (*) verwendet werden, um alle Operationen oder eine Teilmenge von Operationen anzugeben. Beispielsweise entspricht diese Action Berechtigungen wie s3:GetObject, s3:PutObject und s3:DeleteObject.

```
s3:*Object
```

Im Resource-Element können Sie die Platzhalterzeichen (*) und (?) verwenden. Während das Sternchen (*) mit 0 oder mehr Zeichen übereinstimmt, stimmt das Fragezeichen (?) mit einem einzelnen beliebigen Zeichen überein.

Im Principal-Element werden Platzhalterzeichen nicht unterstützt, außer um anonymer Zugriff zu setzen, der allen Benutzern die Berechtigung erteilt. Zum Beispiel wird das Platzhalterzeichen (*) als Wert für Principal gesetzt.

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

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

Im folgenden Beispiel wird die Anweisung mit den Elementen „Effect“, „Principal“, „Action“ und „Resource“ verwendet. Dieses Beispiel zeigt eine vollständige Bucket-Policy-Anweisung, die den Effect „Allow“ verwendet, um den Principals, der admin group `federated-group/admin` und der finance group `federated-group/finance`, Berechtigungen zu erteilen, die Action `s3:ListBucket` für den Bucket mit dem Namen `mybucket` und die Action `s3:GetObject` für alle Objekte in diesem Bucket auszuführen.

```
{
  "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 hat eine Größenbeschränkung von 5.120 Bytes.

Einheitlichkeit der Richtlinien

Standardmäßig sind alle Aktualisierungen, die Sie an Gruppenrichtlinien vornehmen, letztendlich konsistent. Wenn eine Gruppenrichtlinie konsistent ist, kann es aufgrund des Richtlinien-Cachings bis zu 15 Minuten dauern, bis die Änderungen wirksam werden. Standardmäßig sind alle Aktualisierungen, die Sie an Bucket-Richtlinien vornehmen, stark konsistent.

Bei Bedarf können die Konsistenzgarantien für Aktualisierungen der Bucket-Richtlinie geändert werden. Beispielsweise kann gewünscht sein, dass eine Änderung an einer Bucket-Richtlinie auch während eines Standortausfalls verfügbar ist.

In diesem Fall kann entweder der `Consistency-Control`-Header in der PutBucketPolicy-Anfrage gesetzt werden oder die PUT Bucket consistency-Anfrage verwendet werden. Wenn eine Bucket-Richtlinie konsistent wird, können die Änderungen aufgrund des Richtlinien-Cachings bis zu 8 Sekunden benötigen, um wirksam zu werden.

Wenn die Konsistenz vorübergehend auf einen anderen Wert gesetzt wird, um eine temporäre Situation zu umgehen, sollte die Einstellung auf Bucket-Ebene nach Abschluss wieder auf den ursprünglichen Wert zurückgesetzt werden. Andernfalls werden alle zukünftigen Bucket-Anfragen die geänderte Einstellung verwenden.

Was ist eine Sitzungsrichtlinie?

Eine Sitzungsrichtlinie ist eine Zugriffsrichtlinie, die die während einer bestimmten Sitzung verfügbaren Berechtigungen vorübergehend einschränkt, beispielsweise wenn ein Benutzer einer Gruppe beitritt. Eine Sitzungsrichtlinie kann nur eine Teilmenge der Berechtigungen zulassen und keine zusätzlichen Berechtigungen gewähren. Die Gruppe selbst kann über umfassendere Berechtigungen verfügen.

ARN in Richtlinienaussagen verwenden

In Richtlinienanweisungen wird der ARN in Principal- und Resource-Elementen verwendet.

- Diese Syntax gibt den S3-Ressourcen-ARN an:

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

- Diese Syntax gibt den ARN der Identitätsressource (Benutzer und Gruppen) an:

```
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 innerhalb des Objektschlüssels abzugleichen.
- Internationale Zeichen, die im Objektschlüssel angegeben werden können, sollten mit JSON UTF-8 oder mit JSON \u-Escape-Sequenzen kodiert werden. Prozentkodierung wird nicht unterstützt.

  "RFC 2141 URN-Syntax"

  Der HTTP-Anfragetext für die PutBucketPolicy-Operation muss mit charset=UTF-8 kodiert sein.

Ressourcen in einer Richtlinie angeben

In Richtlinienanweisungen kann das Resource-Element verwendet werden, um den Bucket oder das Objekt anzugeben, für das Berechtigungen erlaubt oder verweigert werden.

- Jede Richtlinienaussage erfordert ein Ressourcenelement. In einer Richtlinie werden Ressourcen durch das Element `Resource` oder alternativ `NotResource` für Ausschluss gekennzeichnet.

- Sie geben Ressourcen mithilfe eines S3-Ressourcen-ARN an. Zum Beispiel:

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

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

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

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

Geben Sie Prinzipale in einer Richtlinie an.

Das Principal-Element dient zur Identifizierung des Benutzers, der Gruppe oder des Mandantenkontos, dem der Zugriff auf die Ressource durch die Richtlinienanweisung erlaubt oder 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 Principal gilt.
- In einer Richtlinie werden Principals durch das Element „Principal“ oder alternativ „NotPrincipal“ für den Ausschluss bezeichnet.
- Kontobasierte Identitäten müssen mithilfe einer ID oder einer ARN angegeben werden:

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

- Dieses Beispiel verwendet die Mandantenkonto-ID 27233906934684427525, die das Stammkonto und alle Benutzer des Kontos 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 („Managers“) angeben:

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

- Es kann ein anonymer Prinzipal angegeben werden:

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

- Um Mehrdeutigkeiten zu vermeiden, kann die Benutzer-UUID anstelle des Benutzernamens verwendet werden:

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

  Angenommen, Alex verlässt das Unternehmen und der Benutzername `Alex` wird gelöscht. Wenn ein neuer Alex dem Unternehmen beitritt und denselben `Alex` Benutzernamen erhält, könnte der neue Benutzer unbeabsichtigt die Berechtigungen des ursprünglichen Benutzers übernehmen.

- Der Principal-Wert kann einen Gruppen-/Benutzernamen angeben, der beim Erstellen einer Bucket-Richtlinie noch nicht existiert.

Berechtigungen in einer Richtlinie angeben

In einer Richtlinie wird das Element „Action“ verwendet, um Berechtigungen für eine Ressource zu erlauben oder zu verweigern. Es gibt eine Reihe von Berechtigungen, die in einer Richtlinie angegeben werden können, die durch das Element „Action“ oder alternativ „NotAction“ für den Ausschluss gekennzeichnet sind. Jedes dieser Elemente entspricht bestimmten S3 REST API-Operationen.

Die Tabelle listet die Berechtigungen auf, die für Buckets gelten, und die Berechtigungen, die für Objekte gelten.

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

> Ein Löschvorgang wird durchgeführt, wenn mit einem Put ein vorhandener Wert überschrieben wird.

Berechtigungen, die für Buckets gelten

Berechtigungen	S3 REST API Operationen	Kundenspezifisch für StorageGRID
s3:CreateBucket	CreateBucket	Ja. **Hinweis**: Nur in Gruppenrichtlinien verwenden.

Berechtigungen	S3 REST API Operationen	Kundenspezifisch für StorageGRID
s3:DeleteBucket	DeleteBucket	
s3:DeleteBucketMetadataNotification	Bucket-Metadatenbenachrichtigungskonfiguration löschen	Ja
s3:DeleteBucketPolicy	DeleteBucketPolicy	
s3:DeleteReplicationConfiguration	DeleteBucketReplication	Ja, separate Berechtigungen für PUT und DELETE
s3:GetBucketAcl	GetBucketAcl	
s3:GetBucketCompliance	GET Bucket compliance (veraltet)	Ja
s3:GetBucketConsistency	GET Bucket Konsistenz	Ja
s3:GetBucketCORS	GetBucketCors	
s3:GetEncryptionConfiguration	GetBucketEncryption	
s3:GetBucketLastAccessTime	Letzte Zugriffszeit des Buckets abrufen	Ja
s3:GetBucketLocation	GetBucketLocation	
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	

Berechtigungen	S3 REST API Operationen	Kundenspezifisch für StorageGRID
s3:ListAllMyBuckets	• ListBuckets • Speichernutzung abrufen	Ja, für GET Storage Usage. **Hinweis**: Nur in Gruppenrichtlinien verwenden.
s3:ListBucket	• ListObjects • HeadBucket • RestoreObject	
s3:ListBucketMultipartUploads	• ListMultipartUploads • RestoreObject	
s3:ListBucketVersions	GET Bucket-Versionen	
s3:PutBucketCompliance	PUT Bucket Compliance (veraltet)	Ja
s3:PutBucketConsistency	PUT Bucket Konsistenz	Ja
s3:PutBucketCORS	• DeleteBucketCors† • PutBucketCors	
s3:PutEncryptionConfiguration	• DeleteBucketEncryption • PutBucketEncryption	
s3:PutBucketLastAccessTime	PUT Bucket letzter Zugriffszeitpunkt	Ja
s3:PutBucketMetadataNotification	PUT Bucket Metadaten-Benachrichtigungskonfiguration	Ja
s3:PutBucketNotification	PutBucketNotificationConfiguration	
s3:PutBucketObjectLockConfiguration	• CreateBucket mit dem `x-amz-bucket-object-lock-enabled: true` Request-Header (erfordert außerdem die Berechtigung s3:CreateBucket) • PutObjectLockConfiguration	
s3:PutBucketPolicy	PutBucketPolicy	

Berechtigungen	S3 REST API Operationen	Kundenspezifisch für StorageGRID
s3:PutBucketTagging	• DeleteBucketTagging† • PutBucketTagging	
s3:PutBucketVersioning	PutBucketVersioning	
s3:PutLifecycleConfiguration	• DeleteBucketLifecycle† • PutBucketLifecycleConfiguration	
s3:PutReplicationConfiguration	PutBucketReplication	Ja, separate Berechtigungen für PUT und DELETE

Berechtigungen, die für Objekte gelten

Berechtigungen	S3 REST API Operationen	Kundenspezifisch für StorageGRID
s3:AbortMultipartUpload	• AbortMultipartUpload • RestoreObject	
s3:BypassGovernanceRetention	• DeleteObject • DeleteObjects • PutObjectRetention	
s3:DeleteObject	• DeleteObject • DeleteObjects • 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	

Berechtigungen	S3 REST API Operationen	Kundenspezifisch für StorageGRID
s3:GetObjectAcl	GetObjectAcl	
s3:GetObjectLegalHold	GetObjectLegalHold	
s3:GetObjectRetention	GetObjectRetention	
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 • CopyObject • 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 • CopyObject • PutObjectTagging • DeleteObjectTagging • CompleteMultipartUpload	Ja
s3:RestoreObject	RestoreObject	

PutOverwriteObject-Berechtigung verwenden

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

Mögliche Einstellungen für diese Berechtigung sind:

- **Zulassen**: Der Client kann ein Objekt überschreiben. Dies ist die Standardeinstellung.
- **Verweigern**: Der Client kann ein Objekt nicht überschreiben. Wenn auf „Verweigern“ gesetzt, funktioniert die PutOverwriteObject-Berechtigung wie folgt:
  - Wenn unter demselben Pfad bereits ein Objekt gefunden wird:
    - Die Daten des Objekts, die benutzerdefinierten Metadaten oder die S3-Objektkennzeichnung können nicht überschrieben werden.
    - Alle laufenden Datenerfassungsvorgänge werden abgebrochen und ein Fehler zurückgegeben.
    - Wenn die S3-Versionierung aktiviert ist, verhindert die Einstellung „Deny“, dass PutObjectTagging- oder DeleteObjectTagging-Operationen das TagSet eines Objekts und seiner nicht aktuellen Versionen ändern.
  - Wenn kein vorhandenes Objekt gefunden wird, hat diese Berechtigung keine Auswirkung.
- Wenn diese Berechtigung nicht vorhanden ist, ist die Wirkung identisch damit, als wäre „Zulassen“ gesetzt.

> Wenn die aktuelle S3-Richtlinie das Überschreiben zulässt und die PutOverwriteObject-Berechtigung auf Verweigern gesetzt ist, kann der Client weder die Daten eines Objekts, benutzerdefinierte Metadaten noch Objekt-Tags überschreiben. Wenn außerdem das Kontrollkästchen **Prevent client modification** aktiviert ist (**Configuration** > **Security settings** > **Network and objects**), hat diese Einstellung Vorrang vor der Einstellung der PutOverwriteObject-Berechtigung.

Bedingungen in einer Richtlinie angeben

Bedingungen definieren, wann eine Richtlinie wirksam wird. 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:

- Zeichenkette
- Numerisch
- Boolescher Wert
- IP-Adresse
- Nullprüfung

Bedingungsoperatoren	Beschreibung
StringEquals	Vergleicht einen Schlüssel mit einem Zeichenkettenwert auf Basis einer exakten Übereinstimmung (Groß-/Kleinschreibung).
StringNotEquals	Vergleicht einen Schlüssel mit einem Zeichenkettenwert anhand einer negierten Übereinstimmung (Groß-/Kleinschreibung).
StringEqualsIgnoreCase	Vergleicht einen Schlüssel mit einem Zeichenkettenwert anhand einer exakten Übereinstimmung (Groß-/Kleinschreibung wird ignoriert).
StringNotEqualsIgnoreCase	Vergleicht einen Schlüssel mit einem Zeichenkettenwert auf Basis einer negierten Übereinstimmung (Groß-/Kleinschreibung wird ignoriert).
StringLike	Vergleicht einen Schlüssel mit einem Zeichenkettenwert anhand einer exakten Übereinstimmung (Groß-/Kleinschreibung). Kann die Platzhalterzeichen * und ? enthalten.
StringNotLike	Vergleicht einen Schlüssel mit einem Zeichenkettenwert anhand negierter Übereinstimmung (Groß-/Kleinschreibung). Kann die Platzhalterzeichen * und ? enthalten.
NumericEquals	Vergleicht einen Schlüssel mit einem numerischen Wert basierend auf exakter Übereinstimmung.
NumericNotEquals	Vergleicht einen Schlüssel mit einem numerischen Wert auf Basis einer negierten Übereinstimmung.

Bedingungsoperatoren	Beschreibung
NumericGreaterThan	Vergleicht einen Schlüssel mit einem numerischen Wert auf Basis einer „größer als“-Übereinstimmung.
NumericGreaterThanEquals	Vergleicht einen Schlüssel mit einem numerischen Wert anhand einer Übereinstimmung nach dem Prinzip „größer als oder gleich“.
NumericLessThan	Vergleicht einen Schlüssel mit einem numerischen Wert auf Basis einer „kleiner als“-Übereinstimmung.
NumericLessThanEquals	Vergleicht einen Schlüssel mit einem numerischen Wert auf Basis einer „kleiner als oder gleich“-Übereinstimmung.
Bool	Vergleicht einen Schlüssel mit einem booleschen Wert basierend auf „wahr“ oder „falsch“ Übereinstimmung.
IpAddress	Vergleicht einen Schlüssel mit einer IP-Adresse oder einem Bereich von IP-Adressen.
NotIpAddress	Vergleicht einen Schlüssel mit einer IP-Adresse oder einem Bereich von IP-Adressen auf Basis einer negierten Übereinstimmung.
Null	Prüft, ob ein Bedingungsschlüssel im aktuellen Anfragekontext vorhanden ist.
IfExists	Wird an jeden Bedingungsoperator außer der Null-Bedingung angehängt, um das Fehlen des entsprechenden Bedingungsschlüssels zu prüfen. Gibt TRUE zurück, wenn der Bedingungsschlüssel nicht vorhanden ist.

Unterstützte Bedingungsschlüssel

Bedingungsschlüssel	Aktionen	Beschreibung
aws:SourceIp	IP-Operatoren	Wird mit der IP-Adresse verglichen, von der die Anfrage gesendet wurde. Kann für Bucket- oder Objektoperationen verwendet werden. **Hinweis:** Wenn die S3-Anfrage über den Load Balancer Service auf Admin Nodes und Gateways Nodes gesendet wurde, erfolgt der Vergleich mit der IP-Adresse, die stromaufwärts des Load Balancer Service liegt. **Hinweis**: Bei Verwendung eines nicht transparenten Load Balancers eines Drittanbieters erfolgt ein Vergleich mit der IP-Adresse dieses Load Balancers. Sämtliche `X-Forwarded-For` Header werden ignoriert, da ihre Gültigkeit nicht überprüft 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 im Parameter Trennzeichen einer ListObjects- oder ListObjectVersions-Anfrage angegebenen Wert verglichen.

Bedingungsschlüssel	Aktionen	Beschreibung
s3:ExistingObjectTag/<tag-key>	s3:DeleteObjectTagging s3:DeleteObjectVersionTagging s3:GetObject s3:GetObjectAcl 3:GetObjectTagging s3:GetObjectVersion s3:GetObjectVersionAcl s3:GetObjectVersionTagging s3:PutObjectAcl s3:PutObjectTagging s3:PutObjectVersionAcl s3:PutObjectVersionTagging	Es wird vorausgesetzt, dass das bestehende Objekt den spezifischen Tag-Schlüssel und Wert besitzt.
s3:max-keys	s3:ListBucket und s3:ListBucketVersions-Berechtigungen	Wird mit dem in einer ListObjects oder ListObjectVersions Anfrage angegebenen Parameter max-keys verglichen.
s3:object-lock-mode	s3:PutObject	Vergleicht mit dem `object-lock-mode` erweiterten Anfrageheader in der PutObject, CopyObject und CreateMultipartUpload Anfrage.
s3:object-lock-mode	s3:PutObjectRetention	Vergleicht mit dem `object-lock-mode` aus dem XML-Body in der PutObjectRetention Anfrage erweiterten Inhalt.
s3:object-lock-remaining-retention-days	s3:PutObject	Vergleicht das im `x-amz-object-lock-retain-until-date`Anforderungsheader angegebene oder aus der Standardaufbewahrungsdauer des Buckets berechnete Aufbewahrungsdatum, um sicherzustellen, dass diese Werte im zulässigen Bereich für die folgenden Anfragen liegen: • PutObject • CopyObject • CreateMultipartUpload

Bedingungsschlüssel	Aktionen	Beschreibung
s3:object-lock-remaining-retention-days	s3:PutObjectRetention	Vergleicht das in der PutObjectRetention-Anfrage angegebene Aufbewahrungsdatum mit dem Aufbewahrungsfristdatum, um sicherzustellen, dass es innerhalb des zulässigen Bereichs liegt.
s3:prefix	s3:ListBucket und s3:ListBucketVersions-Berechtigungen	Wird mit dem in einer ListObjects oder ListObjectVersions Anfrage angegebenen Präfixparameter verglichen.
s3:RequestObjectTag/<tag-key>	s3:PutObject s3:PutObjectTagging s3:PutObjectVersionTagging	Wenn die Objektanfrage Tags enthält, sind ein spezifischer Tag-Schlüssel und ein spezifischer Tag-Wert erforderlich.
s3:x-amz-server-side-encryption-customer-algorithm	s3:PutObject	Vergleicht mit dem `sse-customer-algorithm` oder mit dem `copy-source-sse-customer-algorithm` aus dem Anfrage-Header, der in der PutObject, CopyObject, CreateMultipartUpload, UploadPart, UploadPartCopy und CompleteMultipartUpload Anfrage erweitert wurde.

Variablen in einer Richtlinie angeben

Sie können Variablen in Richtlinien verwenden, um Richtlinieninformationen einzufügen, sobald diese verfügbar sind. Sie können Richtlinienvariablen im `Resource` Element und bei Zeichenkettenvergleichen im `Condition` Element verwenden.

In diesem Beispiel ist die Variable `${aws:username}` Teil des Resource-Elements:

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

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

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

Variable	Beschreibung
`${aws:SourceIp}`	Verwendet den SourceIp-Schlüssel als die bereitgestellte Variable.
`${aws:username}`	Verwendet den Schlüssel für den Benutzernamen als bereitgestellte Variable.
`${s3:prefix}`	Verwendet den dienstspezifischen Präfixschlüssel als bereitgestellte Variable.
`${s3:max-keys}`	Verwendet den dienstspezifischen Schlüssel max-keys als bereitgestellte Variable.
`${*}`	Sonderzeichen. Verwendet das Zeichen als wörtliches * Zeichen.
`${?}`	Sonderzeichen. Verwendet das Zeichen als wörtliches Fragezeichen.
`${$}`	Sonderzeichen. Verwendet das Zeichen als wörtliches $-Zeichen.

Richtlinien erstellen, die eine besondere Behandlung erfordern

Richtlinien können mitunter Berechtigungen erteilen, die sicherheitsgefährdend oder für den laufenden Betrieb problematisch sind, beispielsweise die Sperrung des Root-Benutzers des Kontos. Die StorageGRID S3 REST API Implementierung ist bei der Richtlinienvalidierung weniger restriktiv als Amazon, bei der Richtlinienauswertung jedoch genauso streng.

Richtlinienbeschreibung	Richtlinientyp	Amazon Verhalten	StorageGRID Verhalten
Jegliche Berechtigungen für das Root-Konto für sich selbst verweigern	Bucket	Gültig und durchgesetzt, aber das Root-Benutzer-Konto behält die Berechtigung für alle S3-Bucket-Richtlinienoperationen	Dasselbe
Sich selbst alle Berechtigungen für Benutzer/Gruppe verweigern	Gruppe	Gültig und durchgesetzt	Dasselbe
Einer ausländischen Kontengruppe beliebige Berechtigungen erlauben	Bucket	Ungültiger Principal	Gültig, aber die Berechtigungen für alle S3-Bucket-Richtlinienoperationen führen zu einem 405-Fehler (Methode nicht zulässig), wenn sie durch eine Richtlinie erlaubt sind

Richtlinienbeschreibung	Richtlinientyp	Amazon Verhalten	StorageGRID Verhalten
Einem fremden Konto (Root oder Benutzer) beliebige Berechtigungen gewähren	Bucket	Gültig, aber die Berechtigungen für alle S3-Bucket-Richtlinienoperationen führen zu einem 405-Fehler (Methode nicht zulässig), wenn sie durch eine Richtlinie erlaubt sind	Dasselbe
Allen Berechtigungen für alle Aktionen gewähren	Bucket	Gültig, aber für alle S3-Bucket-Richtlinienoperationen wird für das fremde Root-Konto und die Benutzer der Fehler 405 Method Not Allowed zurückgegeben.	Dasselbe
Allen wird die Berechtigung für alle Aktionen verweigert	Bucket	Gültig und durchgesetzt, aber das Root-Benutzer-Konto behält die Berechtigung für alle S3-Bucket-Richtlinienoperationen	Dasselbe
Der Principal ist ein nicht existierender Benutzer oder eine nicht existierende Gruppe	Bucket	Ungültiger Principal	Gültig
Die Ressource ist ein nicht existierender S3-Bucket	Gruppe	Gültig	Dasselbe
Principal ist eine lokale Gruppe	Bucket	Ungültiger Principal	Gültig
Die Richtlinie gewährt einem Nicht-Inhaber-Konto (einschließlich anonymer Konten) Berechtigungen zum Ablegen von Objekten.	Bucket	Gültig. Die Objekte gehören dem Erstellerkonto, und die Bucket-Richtlinie findet keine Anwendung. Das Erstellerkonto muss die Zugriffsberechtigungen für das Objekt mithilfe von Objekt-ACLs erteilen.	Gültig. Objekte gehören dem Bucket-Eigentümerkonto. Bucket-Richtlinie gilt.

Write-once-read-many (WORM)-Schutz

Sie können Write-Once-Read-Many (WORM) Buckets erstellen, um Daten, benutzerdefinierte Objektmetadaten und S3-Objekt-Tags zu schützen. Die WORM-Buckets werden so konfiguriert, dass neue Objekte erstellt werden können und das Überschreiben oder Löschen vorhandener Inhalte verhindert wird. Eine der hier beschriebenen Methoden kann verwendet werden.

Um sicherzustellen, dass Überschreibungen immer verhindert werden, können Sie Folgendes tun:

- Im Grid Manager unter **Konfiguration** > **Sicherheit** > **Sicherheitseinstellungen** > **Netzwerk und Objekte** das Kontrollkästchen **Clientänderungen verhindern** auswählen.
- Die folgenden Regeln und S3-Richtlinien gelten:
  - Eine PutOverwriteObject DENY-Operation zur S3-Richtlinie hinzufügen.
  - Eine DeleteObject DENY-Operation zur S3-Richtlinie hinzufügen.
  - Eine PutObject ALLOW-Operation zur S3-Richtlinie hinzufügen.

> Die Einstellung DeleteObject auf DENY in einer S3-Richtlinie verhindert nicht, dass ILM Objekte löscht, wenn eine Regel wie „null Kopien nach 30 Tagen“ existiert.

> Selbst wenn all diese Regeln und Richtlinien angewendet werden, bieten sie keinen Schutz vor gleichzeitigen Schreibvorgängen (siehe Situation A). Sie schützen jedoch vor sequenziell abgeschlossenen Überschreibungen (siehe Situation B).

Situation A: Gleichzeitige Schreibvorgänge (nicht abgesichert)

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

Situation B: Sequenziell abgeschlossene Überschreibungen (geschützt)

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

Verwandte Informationen

- "Wie StorageGRID ILM Regeln Objekte verwalten"
- "Beispiel-Bucket-Richtlinien"
- "Beispielgruppenrichtlinien"
- "Beispiel einer Sitzungsrichtlinie"
- "Objekte mit ILM verwalten"
- "Ein Mandantenkonto verwenden"_

Beispiel für eine StorageGRID S3 REST API-Sitzungsrichtlinie

Das folgende Beispiel dient zum Erstellen einer StorageGRID Sitzungsrichtlinie.

Beispiel: Eine Sitzungsrichtlinie, die den Abruf von Objekten ermöglicht, einrichten

In diesem Beispiel darf der Sitzungsbenutzer nur Objekte aus bucket1 abrufen. Alle anderen Aktionen sind implizit untersagt, mit Ausnahme von StorageGRID-spezifischen Aktionen, wie der Verwendung der "s3:PutOverwriteObject" Berechtigung. Die Sitzungsrichtlinie kann beim Aufruf der AssumeRole API als JSON-Datei angegeben werden.

```
{
  "Statement": [
    {
      "Action": "s3:GetObject",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::bucket1/*"
    }
  ]
}
```

StorageGRID S3 REST API-Bucket-Richtlinienbeispiele

In diesem Abschnitt finden Sie Beispiele, mit denen StorageGRID Zugriffsrichtlinien für Buckets erstellt werden können.

Bucket-Richtlinien legen die Zugriffsberechtigungen für den Bucket fest, dem die Richtlinie zugeordnet ist. Eine Bucket-Richtlinie wird mithilfe der S3 PutBucketPolicy API über eines der folgenden Tools konfiguriert:

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

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

Beispiel: Allen nur Lesezugriff auf einen Bucket gewähren

In diesem Beispiel darf jeder, einschließlich anonymer Benutzer, Objekte im Bucket auflisten und GetObject-Operationen für alle Objekte im Bucket durchführen. Alle anderen Operationen werden verweigert. Es ist zu beachten, dass diese Richtlinie möglicherweise nicht besonders sinnvoll ist, da außer dem Root-Benutzer des Kontos niemand Berechtigungen zum Schreiben in den Bucket hat.

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

Beispiel: Allen Benutzern eines Kontos Vollzugriff und allen Benutzern eines anderen Kontos nur Lesezugriff auf einen Bucket.

In diesem Beispiel erhält jeder in einem bestimmten Konto vollständigen Zugriff auf einen Bucket, während jeder in einem anderen bestimmten Konto nur berechtigt ist, den Bucket aufzulisten und GetObject-Operationen an Objekten im Bucket durchzuführen, deren Objektschlüssel mit dem `shared/` Objektschlüsselpräfix beginnt.

> In StorageGRID gehören Objekte, die von einem Nicht-Eigentümer-Konto (einschließlich anonymer Konten) erstellt wurden, dem Bucket-Eigentümer-Konto. Die Bucket-Richtlinie gilt für diese Objekte.

```
{
  "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: Allen wird Lesezugriff auf einen Bucket gewährt, während eine bestimmte Gruppe Vollzugriff erhält.

In diesem Beispiel ist es jedem, auch anonymen Benutzern, erlaubt, den Bucket aufzulisten und GetObject-Operationen an allen Objekten im Bucket durchzuführen, während nur Benutzern, die der Gruppe `Marketing` im angegebenen Konto angehören, voller Zugriff gewährt wird.

```
{
  "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: Allen Clients im IP-Bereich Lese- und Schreibzugriff auf einen Bucket erlauben

In diesem Beispiel darf jeder, einschließlich anonymer Benutzer, den Bucket auflisten und beliebige Objektoperationen an allen Objekten im Bucket durchführen, sofern die Anfragen aus einem bestimmten IP-Bereich (54.240.143.0 bis 54.240.143.255, außer 54.240.143.188) stammen. Alle anderen Operationen werden abgelehnt, und alle Anfragen außerhalb dieses 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 einem bestimmten Verbundbenutzer erlauben

In diesem Beispiel hat der Verbundbenutzer Alex vollen Zugriff auf den `examplebucket` Bucket und seine Objekte. Allen anderen Benutzern, einschließlich ‘root’, werden alle Operationen explizit verweigert. Es ist jedoch zu beachten, dass ‘root’ niemals die Berechtigungen für Put/Get/DeleteBucketPolicy entzogen werden.

```
{
  "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 stellt der `Deny` Effect für PutOverwriteObject und DeleteObject sicher, dass niemand die Daten des Objekts, die benutzerdefinierten Metadaten und die S3-Objekt-Tags 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 der StorageGRID S3 REST API

In diesem Abschnitt finden Sie Beispiele, mit denen StorageGRID Zugriffsrichtlinien für Gruppen erstellt werden können.

Gruppenrichtlinien legen die Zugriffsberechtigungen für die Gruppe fest, der die Richtlinie zugeordnet ist. In der Richtlinie ist kein `Principal` Element vorhanden, da es implizit ist. Gruppenrichtlinien werden mit dem Tenant Manager oder der API konfiguriert.

Beispiel: Gruppenrichtlinie mit dem Tenant Manager festlegen

Im Tenant Manager kann beim Hinzufügen oder Bearbeiten einer Gruppe eine Gruppenrichtlinie ausgewählt werden, um festzulegen, über welche S3-Zugriffsberechtigungen die Mitglieder dieser Gruppe verfügen. Siehe "Gruppen für einen S3-Mandanten erstellen".

- **Kein S3-Zugriff**: Standardeinstellung. Benutzer in dieser Gruppe haben keinen Zugriff auf S3-Ressourcen, es sei denn, der Zugriff wird über eine Bucket-Richtlinie gewährt. Bei Auswahl dieser Option hat standardmäßig nur der Root-Benutzer Zugriff auf S3-Ressourcen.
- **Schreibgeschützter Zugriff**: Benutzer in dieser Gruppe haben schreibgeschützten Zugriff auf S3-Ressourcen. Beispielsweise können Benutzer in dieser Gruppe Objekte auflisten sowie Objektdaten, Metadaten und Tags lesen. Bei Auswahl dieser Option erscheint die JSON-Zeichenfolge für eine schreibgeschützte Gruppenrichtlinie im Textfeld. Diese Zeichenfolge kann nicht bearbeitet werden.
- **Vollzugriff**: Benutzer dieser Gruppe haben vollen Zugriff auf S3-Ressourcen, einschließlich Buckets. Bei Auswahl dieser Option wird die JSON-Zeichenfolge für eine Gruppenrichtlinie mit Vollzugriff im Textfeld angezeigt. Diese Zeichenfolge kann nicht bearbeitet werden.
- **Schutz vor Ransomware**: Diese Beispielrichtlinie gilt für alle Buckets dieses Mandanten. Benutzer in dieser Gruppe können allgemeine Aktionen ausführen, jedoch keine Objekte aus Buckets mit aktivierter Objektversionierung dauerhaft löschen.

  Benutzer des Tenant Manager, die über die Berechtigung „Alle Buckets verwalten“ verfügen, können diese Gruppenrichtlinie außer Kraft setzen. Die Berechtigung „Alle Buckets verwalten“ sollte auf vertrauenswürdige Benutzer beschränkt werden, und wo verfügbar, ist Multi-Faktor-Authentifizierung (MFA) zu verwenden.

- **Benutzerdefiniert**: Benutzern in der Gruppe werden die Berechtigungen gewährt, die im Textfeld angegeben werden.

Beispiel: Der Gruppe voller Zugriff auf alle Buckets

In diesem Beispiel haben alle Mitglieder der Gruppe vollen Zugriff auf alle Buckets, die dem Mandantenkonto gehören, es sei denn, dies wird durch die Bucket-Richtlinie ausdrücklich verweigert.

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

Beispiel: Gruppenlesezugriff auf alle Buckets zulassen

In diesem Beispiel haben alle Mitglieder der Gruppe nur Lesezugriff auf S3-Ressourcen, sofern dies nicht ausdrücklich durch die Bucket-Richtlinie verweigert wird. Benutzer in dieser Gruppe können beispielsweise Objekte auflisten sowie 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: Gruppenmitglieder erhalten vollen Zugriff nur auf ihren „Ordner“ in einem Bucket

In diesem Beispiel dürfen Gruppenmitglieder nur ihren jeweiligen Ordner (Schlüsselpräfix) im angegebenen Bucket auflisten und darauf zugreifen. Es ist zu beachten, dass bei der Festlegung der Vertraulichkeit dieser Ordner auch die Zugriffsberechtigungen anderer 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}/*"
    }
  ]
}
```

S3-Operationen, die in den StorageGRID-Audit-Protokollen protokolliert werden

Audit-Meldungen werden von StorageGRID-Diensten generiert und in Textprotokolldateien gespeichert. Im Revisionsprotokoll können die S3-spezifischen Audit-Meldungen überprüft werden, um Details zu Bucket- und Objektoperationen zu erhalten.

Bucket-Operationen werden in den Revisionsprotokollen protokolliert.

- CreateBucket
- DeleteBucket
- DeleteBucketTagging
- DeleteObjects
- GetBucketTagging
- HeadBucket
- ListObjects
- ListObjectVersions
- PUT Bucket Compliance
- PutBucketTagging

- PutBucketVersioning

Objektoperationen, die in den Revisionsprotokollen protokolliert werden

- CompleteMultipartUpload
- CopyObject
- DeleteObject
- GetObject
- HeadObject
- PutObject
- RestoreObject
- SelectObject
- UploadPart (wenn eine ILM-Regel Balanced oder Strict Ingest verwendet)
- UploadPartCopy (wenn eine ILM-Regel Balanced oder Strict für die Datenerfassung verwendet)

Verwandte Informationen

- "Revisionsprotokolldatei aufrufen"
- "Client schreibt Revisionsprotokollmeldungen"
- "Client-Lese-Revisionsprotokollmeldungen"