S3 REST API 사용

StorageGRID software

NetApp
July 23, 2026

목차

S3 REST API 사용

StorageGRID S3 REST API 지원 버전 및 업데이트

StorageGRID REST(Representational State Transfer) 웹 서비스 세트로 구현된 S3(Simple Storage Service) API를 지원합니다.

S3 REST API 지원을 통해 S3 웹 서비스를 위해 개발된 서비스 지향 애플리케이션을 StorageGRID 시스템을 사용하는 온프레미스 객체 스토리지에 연결할 수 있습니다. 클라이언트 애플리케이션에서 현재 S3 REST API 호출을 사용하는 방식에 최소한의 변경만 필요합니다.

지원되는 버전

StorageGRID는 다음과 같은 특정 버전의 STS, S3 및 HTTP를 지원합니다.

항목	버전
STS AssumeRole API 사양	"Amazon Web Services (AWS) 문서: Amazon AssumeRole API 참조" AssumeRole에 대한 자세한 내용은 "AssumeRole 설정"을 참조하십시오.
S3 API 사양	"AWS 문서: Amazon Simple Storage Service API 참조"
HTTP	1.1 HTTP에 대한 자세한 내용은 HTTP/1.1(RFC 7230-35)을 참조하십시오. "IETF RFC 2616: Hypertext Transfer Protocol (HTTP/1.1)" 참고: StorageGRID는 HTTP/1.1 파이프라이닝을 지원하지 않습니다.

S3 REST API 지원 업데이트

릴리스	댓글
12.0	• 관리 인터페이스에 대한 CORS(Cross-Origin Resource Sharing)에 대한 지원이 추가되어 다른 도메인에서 관리 API를 사용하여 StorageGRID의 데이터에 액세스할 수 있습니다. "더 알아보기". • STS AssumeRole 및 세션 정책 지원이 추가되었습니다. "세션 정책의 예시"을 참조하십시오. 테넌트 그룹에서 AssumeRole을 설정할 수 있습니다.

릴리스	댓글
11.9	• 다음 요청 및 지원되는 헤더에 대해 미리 계산된 SHA-256 체크섬 값에 대한 지원이 추가되었습니다. 이 기능을 사용하여 업로드된 객체의 무결성을 확인할 수 있습니다. ◦ 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` • 그리드 관리자가 테넌트 수준의 보존 및 규정 준수 설정을 제어할 수 있는 기능을 추가했습니다. 이러한 설정은 S3 객체 잠금 설정에 영향을 미칩니다. ◦ 버킷 기본 보존 모드 및 객체 보존 모드: 그리드 관리자가 허용하는 경우 Governance 또는 Compliance. ◦ 버킷 기본 보존 기간 및 객체 보존 만료일: 그리드 관리자가 설정한 최대 보존 기간에서 허용하는 값보다 작거나 같아야 합니다. • 향상된 `aws-chunked` 콘텐츠 인코딩 및 스트리밍 `x-amz-content-sha256` 값 지원. 제한 사항: ◦ 존재하는 경우 `chunk-signature`은 선택 사항이며 유효성 검사를 거치지 않습니다. ◦ 존재하는 경우, `x-amz-trailer` 콘텐츠는 무시됩니다
11.8	S3 작업 이름을 "Amazon Web Services(AWS) 문서: Amazon Simple Storage Service API 참조"에 사용된 이름과 일치하도록 업데이트했습니다.
11.7	• 추가됨 "빠른 참조: 지원되는 S3 API 요청". • S3 객체 잠금과 함께 거버넌스 모드를 사용할 수 있도록 지원이 추가되었습니다. • GET Object 및 HEAD Object 요청에 대한 StorageGRID 관련 `x-ntap-sg-cgr-replication-status` 응답 헤더 지원이 추가되었습니다. 이 헤더는 그리드 간 복제를 위한 객체의 복제 상태를 제공합니다. • SelectObjectContent 요청에서 이제 Parquet 객체를 지원합니다.
11.6	• GET Object 및 HEAD Object 요청에서 `partNumber` request 매개변수를 사용할 수 있도록 지원이 추가되었습니다. • S3 Object Lock에 대해 버킷 수준에서 기본 보존 모드 및 기본 보존 기간에 대한 지원이 추가되었습니다. • `s3:object-lock-remaining-retention-days` 정책 조건 키에 대한 지원이 추가되어 객체에 허용되는 보존 기간 범위를 설정할 수 있습니다. • 단일 PUT Object 작업의 최대 권장 크기를 5GiB(5,368,709,120바이트)로 변경했습니다. 5GiB보다 큰 객체가 있는 경우 멀티파트 업로드를 사용하십시오.

릴리스	댓글
11.5	• 버킷 암호화 관리 지원이 추가되었습니다. • S3 객체 잠금 지원이 추가되었으며, 기존의 규정 준수 요청 기능은 더 이상 사용되지 않습니다. • 버전 관리 버킷에서 DELETE Multiple Objects 사용에 대한 지원이 추가되었습니다. • The `Content-MD5` 요청 헤더가 이제 올바르게 지원됩니다.
11.4	• DELETE Bucket tagging, GET Bucket tagging 및 PUT Bucket tagging에 대한 지원이 추가되었습니다. 비용 할당 태그는 지원되지 않습니다. • StorageGRID 11.4에서 생성된 버킷의 경우, 성능 모범 사례를 충족하기 위해 객체 키 이름을 제한할 필요가 더 이상 없습니다. • `s3:ObjectRestore:Post` 이벤트 유형에 대한 버킷 알림 지원이 추가되었습니다. • 이제 AWS 멀티파트 파트의 크기 제한이 적용됩니다. 멀티파트 업로드의 각 파트는 5MiB에서 5GiB 사이여야 합니다. 마지막 파트는 5MiB보다 작을 수 있습니다. • TLS 1.3 지원이 추가되었습니다.
11.3	• 고객이 제공하는 키를 사용한 서버 측 객체 데이터 암호화(SSE-C) 지원이 추가되었습니다. • 버킷 수명 주기 작업(만료 작업에만 해당) 및 `x-amz-expiration` 응답 헤더에 대한 DELETE, GET, PUT Bucket 작업 지원이 추가되었습니다. • PUT Object, PUT Object - Copy 및 Multipart Upload가 업데이트되어 수집 시 동기 배치를 사용하는 ILM 규칙의 영향을 설명합니다. • TLS 1.1 암호화 방식은 더 이상 지원되지 않습니다.
11.2	클라우드 스토리지 풀에서 사용할 수 있도록 POST 객체 복원 지원이 추가되었습니다. 그룹 및 버킷 정책에서 ARN, 정책 조건 키 및 정책 변수에 AWS 구문을 사용할 수 있도록 지원이 추가되었습니다. StorageGRID 구문을 사용하는 기존 그룹 및 버킷 정책은 계속 지원됩니다. 참고**:** 사용자 지정 StorageGRID 기능에 사용되는 것을 포함하여 다른 구성 JSON/XML에서 ARN/URN의 사용 방식은 변경되지 않았습니다.
11.1	CORS(교차 출처 리소스 공유) 지원, 그리드 노드에 대한 S3 클라이언트 연결용 HTTP 지원, 버킷의 규정 준수 설정 기능이 추가되었습니다.
11.0	버킷에 대한 플랫폼 서비스(CloudMirror 복제, 알림 및 Elasticsearch 검색 통합) 구성 지원이 추가되었습니다. 또한 버킷에 대한 객체 태깅 위치 제약 조건 및 Available 일관성 지원도 추가되었습니다.
10.4	버전 관리, 엔드포인트 도메인 이름 페이지 업데이트, 정책의 조건 및 변수, 정책 예제, PutOverwriteObject 권한에 대한 ILM 스캔 변경 사항에 대한 지원이 추가되었습니다.
10.3	버전 관리 기능이 추가되었습니다.

릴리스	댓글
10.2	그룹 및 버킷 액세스 정책과 멀티파트 복사(Upload Part - Copy)에 대한 지원이 추가되었습니다.
10.1	멀티파트 업로드, 가상 호스팅 방식 요청 및 v4 인증에 대한 지원이 추가되었습니다.
10.0	StorageGRID 시스템에서 S3 REST API를 처음으로 지원합니다. 현재 지원되는 *Simple Storage Service API Reference* 버전은 2006-03-01입니다.

StorageGRID에서 지원되는 S3 API 요청

이 페이지에서는 StorageGRID가 Amazon Simple Storage Service(S3) API를 지원하는 방법을 요약합니다.

이 페이지에는 StorageGRID에서 지원하는 S3 작업만 포함되어 있습니다.

각 작업에 대한 AWS 설명서를 보려면 제목의 링크를 선택하십시오.

일반적인 URI 쿼리 매개변수 및 요청 헤더

별도로 명시되지 않은 경우, 다음과 같은 일반적인 URI 쿼리 매개변수가 지원됩니다.

- `versionId` (객체 연산에 필요한 사항)

별도로 명시되지 않은 경우, 다음과 같은 일반 요청 헤더가 지원됩니다.

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

관련 정보

- "S3 REST API 구현 세부 정보"
- "Amazon Simple Storage Service API 참조: 일반적인 요청 헤더"

"AbortMultipartUpload"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 을 지원하며, 추가적으로 다음 URI 쿼리 매개변수도 지원합니다.

- `uploadId`

요청 본문

None

StorageGRID 문서

"멀티파트 업로드 작업"

"CompleteMultipartUpload"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 을 지원하며, 추가적으로 다음 URI 쿼리 매개변수도 지원합니다.

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

요청 본문 **XML** 태그

StorageGRID는 다음과 같은 요청 본문 XML 태그를 지원합니다.

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

StorageGRID 문서

"CompleteMultipartUpload"

"CopyObject"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 을 지원하며, 다음과 같은 추가 헤더도 지원합니다.

- `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>`

요청 본문

None

StorageGRID 문서

"CopyObject"

"CreateBucket"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 을 지원하며, 다음과 같은 추가 헤더도 지원합니다.

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

요청 본문

StorageGRID 구현 당시 Amazon S3 REST API에서 정의한 모든 요청 본문 매개변수를 지원합니다.

StorageGRID 문서

"버킷에 대한 작업"

"CreateMultipartUpload"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 을 지원하며, 다음과 같은 추가 헤더도 지원합니다.

- `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>`

요청 본문

None

StorageGRID 문서

"CreateMultipartUpload"

"DeleteBucket"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

StorageGRID 문서

"버킷에 대한 작업"

"DeleteBucketCors"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"DeleteBucketEncryption"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"DeleteBucketLifecycle"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

- "버킷에 대한 작업"
- "S3 라이프사이클 구성 생성"

"DeleteBucketPolicy"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"DeleteBucketReplication"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"DeleteBucketTagging"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"DeleteObject"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더을 지원하며, 추가로 다음 요청 헤더도 지원합니다.

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

요청 본문

None

StorageGRID 문서

"객체에 대한 연산"

"DeleteObjects"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더을 지원하며, 추가로 다음 요청 헤더도 지원합니다.

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

요청 본문

StorageGRID 구현 당시 Amazon S3 REST API에서 정의한 모든 요청 본문 매개변수를 지원합니다.

StorageGRID 문서

"객체에 대한 연산"

"DeleteObjectTagging"

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"객체에 대한 연산"

"GetBucketAcl"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketCors"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketEncryption"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketLifecycleConfiguration"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

- "버킷에 대한 작업"
- "S3 라이프사이클 구성 생성"

"GetBucketLocation"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketNotificationConfiguration"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketPolicy"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketReplication"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketTagging"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetBucketVersioning"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"GetObject"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 URI 쿼리 매개변수를 지원하며, 추가로 다음과 같은 URI 쿼리 매개변수도 지원합니다.

- `x-amz-checksum-mode`
- `partNumber`
- `response-cache-control`
- `response-content-disposition`
- `response-content-encoding`
- `response-content-language`
- `response-content-type`
- `response-expires`

그리고 다음과 같은 추가 요청 헤더가 있습니다.

- `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`

요청 본문

None

StorageGRID 문서

"GetObject"

"GetObjectAcl"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"객체에 대한 연산"

"GetObjectLegalHold"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"S3 REST API를 사용하여 S3 Object Lock 구성"

"GetObjectLockConfiguration"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"S3 REST API를 사용하여 S3 Object Lock 구성"

"GetObjectRetention"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"S3 REST API를 사용하여 S3 Object Lock 구성"

"GetObjectTagging"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"객체에 대한 연산"

"HeadBucket"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"HeadObject"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 을 지원하며, 다음과 같은 추가 헤더도 지원합니다.

- `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`

요청 본문
None

StorageGRID 문서
"HeadObject"

"ListBuckets"

URI 쿼리 매개변수 및 요청 헤더
StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문
None

StorageGRID 문서
서비스 › **ListBuckets**에 대한 작업

"ListMultipartUploads"

URI 쿼리 매개변수 및 요청 헤더
StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 을 지원하며, 다음과 같은 추가 매개변수도 지원합니다.

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

요청 본문
None

StorageGRID 문서
"ListMultipartUploads"

"ListObjects"

URI 쿼리 매개변수 및 요청 헤더
StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 을 지원하며, 다음과 같은 추가 매개변수도 지원합니다.

- `delimiter`
- `encoding-type`
- `marker`

- `max-keys`
- `prefix`

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"ListObjectsV2"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 을 지원하며, 다음과 같은 추가 매개변수도 지원합니다.

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

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"ListObjectVersions"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 을 지원하며, 다음과 같은 추가 매개변수도 지원합니다.

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

요청 본문

None

StorageGRID 문서

"버킷에 대한 작업"

"ListParts"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 을 지원하며, 다음과 같은 추가 매개변수도 지원합니다.

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

요청 본문

None

StorageGRID 문서

"ListMultipartUploads"

"PutBucketCors"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

StorageGRID 구현 당시 Amazon S3 REST API에서 정의한 모든 요청 본문 매개변수를 지원합니다.

StorageGRID 문서

"버킷에 대한 작업"

"PutBucketEncryption"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문 **XML** 태그

StorageGRID는 다음과 같은 요청 본문 XML 태그를 지원합니다.

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

StorageGRID 문서

"버킷에 대한 작업"

"PutBucketLifecycleConfiguration"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문 **XML** 태그

StorageGRID는 다음과 같은 요청 본문 XML 태그를 지원합니다.

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

StorageGRID 문서

- "버킷에 대한 작업"
- "S3 라이프사이클 구성 생성"

"PutBucketNotificationConfiguration"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문 **XML** 태그

StorageGRID는 다음과 같은 요청 본문 XML 태그를 지원합니다.

- `Event`
- `Filter`
- `FilterRule`

- `Id`
- `Name`
- `NotificationConfiguration`
- `Prefix`
- `S3Key`
- `Suffix`
- `Topic`
- `TopicConfiguration`
- `Value`

StorageGRID 문서

"버킷에 대한 작업"

"PutBucketPolicy"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

지원되는 JSON 본문 필드에 대한 자세한 내용은 "버킷 및 그룹 액세스 정책 사용"을 참조하세요.

"PutBucketReplication"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문 **XML** 태그

- `Bucket`
- `Destination`
- `Prefix`
- `ReplicationConfiguration`
- `Rule`
- `Status`
- `StorageClass`

StorageGRID 문서

"버킷에 대한 작업"

"PutBucketTagging"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

StorageGRID 구현 당시 Amazon S3 REST API에서 정의한 모든 요청 본문 매개변수를 지원합니다.

StorageGRID 문서

"버킷에 대한 작업"

"PutBucketVersioning"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문 매개변수

StorageGRID는 다음과 같은 요청 본문 매개변수를 지원합니다.

- `VersioningConfiguration`
- `Status`

StorageGRID 문서

"버킷에 대한 작업"

"PutObject"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 을 지원하며, 다음과 같은 추가 헤더도 지원합니다.

- `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>`

요청 본문

- 객체의 이진 데이터

StorageGRID 문서

"PutObject"

"PutObjectLegalHold"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

StorageGRID 구현 당시 Amazon S3 REST API에서 정의한 모든 요청 본문 매개변수를 지원합니다.

StorageGRID 문서

"S3 REST API를 사용하여 S3 Object Lock 구성"

"PutObjectLockConfiguration"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

StorageGRID 구현 당시 Amazon S3 REST API에서 정의한 모든 요청 본문 매개변수를 지원합니다.

StorageGRID 문서

"S3 REST API를 사용하여 S3 Object Lock 구성"

"PutObjectRetention"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 헤더를 지원하며, 추가적으로 다음 헤더도 지원합니다.

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

요청 본문

StorageGRID 구현 당시 Amazon S3 REST API에서 정의한 모든 요청 본문 매개변수를 지원합니다.

StorageGRID 문서

"S3 REST API를 사용하여 S3 Object Lock 구성"

"PutObjectTagging"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

StorageGRID 구현 당시 Amazon S3 REST API에서 정의한 모든 요청 본문 매개변수를 지원합니다.

StorageGRID 문서

"객체에 대한 연산"

"RestoreObject"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

지원되는 본문 필드에 대한 자세한 내용은 "RestoreObject"을 참조하십시오.

"SelectObjectContent"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대해 모든 공통 매개변수 및 헤더을 지원합니다.

요청 본문

지원되는 본문 필드에 대한 자세한 내용은 다음을 참조하십시오.

- "S3 Select 사용"
- "SelectObjectContent"

"UploadPart"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 URI 쿼리 매개변수를 지원하며, 추가로 다음과 같은 URI 쿼리 매개변수도 지원합니다.

- `partNumber`
- `uploadId`

그리고 다음과 같은 추가 요청 헤더가 있습니다.

- `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`

요청 본문

- 해당 부분의 이진 데이터

StorageGRID 문서

"UploadPart"

"UploadPartCopy"

URI 쿼리 매개변수 및 요청 헤더

StorageGRID는 이 요청에 대한 모든 공통 매개변수 및 헤더 URI 쿼리 매개변수를 지원하며, 추가로 다음과 같은 URI 쿼리 매개변수도 지원합니다.

- `partNumber`
- `uploadId`

그리고 다음과 같은 추가 요청 헤더가 있습니다.

- `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`

요청 본문

None

StorageGRID 문서

"UploadPartCopy"

StorageGRID S3 REST API 구성 테스트

AWS CLI(Amazon Web Services Command Line Interface)를 사용하여 시스템 연결을 테스트하고 객체를 읽고 쓸 수 있는지 확인할 수 있습니다.

시작하기 전에

- AWS CLI를 다운로드하여 설치했습니다 "aws.amazon.com/cli".

- 선택적으로 "로드 밸런서 엔드포인트를 생성했습니다."이(가) 있습니다. 그렇지 않으면 연결하려는 스토리지 노드의 IP 주소와 사용할 포트 번호를 알고 있어야 합니다. "클라이언트 연결을 위한 IP 주소 및 포트"을 참조하십시오.
- "S3 테넌트 계정을 생성했습니다."이(가) 있습니다.
- 테넌트에 로그인하여 "액세스 키를 생성했습니다"했습니다.

이러한 단계에 대한 자세한 내용은 "클라이언트 연결 구성"을 참조하십시오.

단계

1. StorageGRID 시스템에서 생성한 계정을 사용하도록 AWS CLI 설정을 구성하십시오.
   a. 구성 모드로 진입합니다: `aws configure`
   b. 생성한 계정의 액세스 키 ID를 입력합니다.
   c. 생성한 계정의 비밀 액세스 키를 입력하십시오.
   d. 사용할 기본 지역을 입력하세요. 예를 들어, `us-east-1`.
   e. 사용할 기본 출력 형식을 입력하거나 **Enter** 키를 눌러 JSON을 선택하십시오.
2. 버킷을 생성합니다.

   이 예제에서는 로드 밸런서 엔드포인트가 IP 주소 10.96.101.17 및 포트 10443을 사용하도록 구성되었다고 가정합니다.

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

   버킷 생성이 성공적으로 완료되면 다음 예시와 같이 버킷의 위치가 반환됩니다.

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

3. 객체를 업로드합니다.

   ```
   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
   ```

   객체 업로드가 성공적으로 완료되면 객체 데이터의 해시값인 Etag가 반환됩니다.

4. 버킷의 내용을 나열하여 객체가 업로드되었는지 확인하십시오.

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

5. 개체를 삭제합니다.

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

6. 버킷을 삭제합니다.

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

StorageGRID의 S3 REST API 구현 방식

StorageGRID S3 REST API가 충돌하는 클라이언트 요청을 해결하는 방법

두 클라이언트가 동일한 키에 쓰기를 시도하는 등 클라이언트 요청이 충돌하는 경우, "최신 요청 우선" 원칙에 따라 해결됩니다.

"최신 성공" 평가 시점은 S3 클라이언트가 작업을 시작한 시점이 아니라 StorageGRID 시스템이 특정 요청을 완료한 시점을 기준으로 합니다.

StorageGRID S3 REST API가 가용성과 일관성의 균형을 유지하는 방법

일관성은 객체의 가용성과 다양한 스토리지 노드 및 사이트 간의 객체 일관성 사이의 균형을 제공합니다. 애플리케이션 요구 사항에 따라 일관성 수준을 변경할 수 있습니다.

기본적으로 StorageGRID는 새로 생성된 객체에 대해 쓰기 후 읽기 일관성을 보장합니다. PUT 작업이 성공적으로 완료된 후 발생하는 모든 GET 요청은 새로 기록된 데이터를 읽을 수 있습니다. 기존 객체 덮어쓰기, 메타데이터 업데이트 및 삭제는 최종 일관성을 보장합니다.

객체 작업을 다른 일관성 수준으로 수행하려면 다음을 수행할 수 있습니다.

- 각 버킷에 대한 일관성을 지정합니다.
- 각 API 작업에 대한 일관성을 지정합니다.
- 다음 작업 중 하나를 수행하여 그리드 전체의 기본 일관성 설정을 변경하십시오.
  - 그리드 관리자에서 구성 > 시스템 > 저장소 설정 > *기본 버킷 일관성*으로 이동합니다.
  - .

> 그리드 전체 일관성 설정 변경은 해당 설정이 변경된 후에 생성된 버킷에만 적용됩니다. 변경 사항에 대한 자세한 내용을 확인하려면 `/var/local/log`에 있는 감사 로그를 참조하십시오 (**consistencyLevel** 검색).

일관성 값

일관성은 StorageGRID가 객체를 추적하는 데 사용하는 메타데이터가 노드 간에 분산되는 방식에 영향을 미칩니다. 일관성은 클라이언트 요청에 대한 객체 가용성에 영향을 미칩니다.

버킷 또는 API 작업에 대한 일관성 수준을 다음 값 중 하나로 설정할 수 있습니다.

- 모두: 모든 노드가 객체 메타데이터를 즉시 수신해야 하며, 그렇지 않으면 요청이 실패합니다.
- **Strong-global**: 모든 사이트에서 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다. 쿼럼 시맨틱이 구성된 경우 다음과 같은 동작이 적용됩니다.
  - 그리드에 사이트가 세 개 이상 있는 경우 클라이언트 요청에 대한 사이트 장애 허용 기능을 제공합니다. 사이트가 두 개인 그리드에는 사이트 장애 허용 기능이 없습니다.
  - 다음 S3 작업은 사이트 중 하나라도 다운되면 성공하지 못합니다.
    - DeleteBucketEncryption
    - PutBucketBranch
    - PutBucketEncryption
    - PutBucketVersioning
    - PutObjectLegalHold
    - PutObjectLockConfiguration
    - PutObjectRetention

  필요한 경우 "강력한 전역 일관성을 위해 StorageGRID 쿼럼 시맨틱 구성".
- **Strong-site**: 객체 메타데이터가 사이트 내 다른 노드로 즉시 배포됩니다. 사이트 내 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다.
- **Read-after-new-write**: 새 객체에 대해서는 쓰기 후 읽기 일관성을 제공하고, 객체 업데이트에 대해서는 최종 일관성을 제공합니다. 높은 가용성과 데이터 보호를 보장합니다. 대부분의 경우에 권장됩니다.
- 사용 가능: 새 객체 생성 및 객체 업데이트 모두에 대해 최종 일관성을 제공합니다. S3 버킷의 경우, 필요한 경우에만 사용하십시오(예: 거의 읽지 않는 로그 값이 포함된 버킷 또는 존재하지 않는 키에 대한 HEAD 또는 GET 작업). S3 FabricPool 버킷에서는 지원되지 않습니다.

"새로운 쓰기 후 읽기" 및 **"사용 가능"** 일관성 사용

HEAD 또는 GET 작업에서 "새로 쓰기 후 읽기" 일관성을 사용하는 경우 StorageGRID는 다음과 같이 여러 단계에 걸쳐 조회를 수행합니다.

- 먼저 낮은 일관성을 사용하여 객체를 찾습니다.
- 해당 조회가 실패하면 강력한 전역 설정에 대한 동작과 동일한 일관성 값에 도달할 때까지 다음 일관성 값에서 조회를 반복합니다.

HEAD 또는 GET 작업에서 "새로 쓰기 후 읽기" 일관성 방식을 사용하지만 객체가 존재하지 않는 경우, 객체 조회는 항상 strong-global 방식과 동일한 일관성 상태로 진행됩니다. 이 일관성 방식은 각 사이트에서 객체 메타데이터의 여러 복사본이 있어야 하므로, 동일 사이트의 스토리지 노드 두 개 이상을 사용할 수 없는 경우 500 내부 서버 오류가 많이 발생할 수 있습니다.

Amazon S3와 유사한 일관성 보장이 필요한 경우가 아니라면, HEAD 및 GET 작업에서 일관성 설정을 "Available"로 지정하여 이러한 오류를 방지할 수 있습니다. HEAD 또는 GET 작업에서 "Available" 일관성을 사용하는 경우, StorageGRID는 최종 일관성만 제공합니다. 실패한 작업에 대해 일관성 수준을 높여 재시도하지 않으므로 객체 메타데이터의 여러 복사본을 사용할 수 있어야 하는 요구 사항이 없습니다.

API 작업에 대한 일관성 지정

개별 API 작업에 대한 일관성 설정을 하려면 해당 작업에서 일관성 값이 지원되어야 하며, 요청 헤더에 일관성을 지정해야 합니다. 이 예제는 GetObject 작업에 대해 일관성을 "Strong-site"로 설정합니다.

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

PutObject 및 GetObject 작업 모두에 동일한 일관성을 유지해야 합니다.

버킷에 대한 일관성 지정

버킷의 일관성을 설정하려면 StorageGRID "PUT 버킷 일관성" 요청을 사용할 수 있습니다. 또는 테넌트 관리자에서 "버킷의 일관성 변경" 할 수 있습니다.

버킷의 일관성을 설정할 때 다음 사항에 유의하십시오.

- 버킷의 일관성 설정은 버킷의 객체 또는 버킷 구성에 대해 수행되는 S3 작업에 사용되는 일관성을 결정합니다. 이는 버킷 자체에 대한 작업에는 영향을 미치지 않습니다.
- 개별 API 작업에 대한 일관성 설정이 버킷의 일관성 설정보다 우선합니다.
- 일반적으로 버킷은 기본 일관성 방식인 "새 쓰기 후 읽기"를 사용해야 합니다. 요청이 제대로 작동하지 않으면 가능하면 애플리케이션 클라이언트의 동작을 변경하십시오. 또는 각 API 요청에 대해 일관성 방식을 지정하도록 클라이언트를 구성하십시오. 버킷 수준에서 일관성 방식을 설정하는 것은 최후의 수단으로만 사용하십시오.

일관성과 ILM 규칙이 상호 작용하여 데이터 보호에 영향을 미치는 방식

일관성 설정과 ILM 규칙 모두 객체 보호 방식에 영향을 미칩니다. 이러한 설정은 서로 상호 작용할 수 있습니다.

예를 들어, 객체 저장 시 사용되는 일관성 수준은 객체 메타데이터의 초기 배치에 영향을 미치며, ILM 규칙에 대해 선택된 수집 동작은 객체 복사본의 초기 배치에 영향을 미칩니다. StorageGRID는 클라이언트 요청을 처리하기 위해 객체의 메타데이터와 데이터 모두에 액세스해야 하므로, 일관성 수준과 수집 동작에 대해 동일한 수준의 보호 방식을 선택하면 초기 데이터 보호가 강화되고 시스템 응답이 더욱 예측 가능해질 수 있습니다.

다음 "수집 옵션"은(는) ILM 규칙에 사용할 수 있습니다.

이중 커밋

StorageGRID는 즉시 객체의 임시 복사본을 생성하고 클라이언트에 성공을 반환합니다. ILM 규칙에 지정된 복사본은 가능한 경우 생성됩니다.

엄격한

ILM 규칙에 명시된 모든 사본이 생성되어야만 클라이언트에 성공이 반환됩니다.

균형 잡힌

StorageGRID 데이터 수집 시 ILM 규칙에 지정된 모든 복사본을 생성하려고 시도합니다. 이것이 불가능한 경우, 임시 복사본을 생성하고 클라이언트에 성공을 반환합니다. ILM 규칙에 지정된 복사본은 가능한 경우 생성됩니다.

일관성 규칙과 **ILM** 규칙이 상호 작용하는 방식의 예시

다음과 같은 ILM 규칙과 일관성 조건을 갖는 3개 사이트 그리드가 있다고 가정해 보겠습니다.

- **ILM** 규칙: 객체 복사본 3개를 생성합니다. 하나는 로컬 사이트에, 나머지 두 개는 각 원격 사이트에 생성합니다. 엄격한 수집 동작을 사용합니다.
- 일관성: Strong-global(오브젝트 메타데이터가 여러 사이트에 즉시 배포됨).

클라이언트가 그리드에 객체를 저장하면 StorageGRID는 객체의 세 가지 복사본을 모두 만들고 메타데이터를 여러 사이트에 배포한 후 클라이언트에 성공 응답을 반환합니다.

객체는 메시지 수집 성공 시점에 손실로부터 완벽하게 보호됩니다. 예를 들어, 수집 직후 로컬 사이트에 장애가 발생하더라도 객체 데이터와 객체 메타데이터의 복사본이 원격 사이트에 여전히 남아 있습니다. 따라서 다른 사이트에서 객체를 완벽하게 복구할 수 있습니다.

동일한 ILM 규칙과 강력한 사이트 일관성을 사용하는 경우, 클라이언트는 객체 데이터가 원격 사이트로 복제된 후 객체 메타데이터가 배포되기 전에 성공 메시지를 수신할 수 있습니다. 이 경우 객체 메타데이터의 보호 수준이 객체 데이터의 보호 수준과 일치하지 않습니다. 수집 직후 로컬 사이트가 손실되면 객체 메타데이터도 손실됩니다. 객체를 검색할 수 없습니다.

일관성과 ILM 규칙 간의 상호 관계는 복잡할 수 있습니다. 도움이 필요하시면 NetApp에 문의하십시오.

StorageGRID에서 객체 버전 관리를 사용하는 방법

버킷에 버전 관리 기능을 설정하면 각 객체의 여러 버전을 보존할 수 있습니다. 버킷에 버전 관리를 활성화하면 객체가 실수로 삭제되는 것을 방지하고 이전 버전의 객체를 검색 및 복원할 수 있습니다.

StorageGRID 시스템은 대부분의 기능을 지원하는 버전 관리 기능을 구현하지만, 몇 가지 제한 사항이 있습니다. StorageGRID는 각 객체에 대해 최대 10,000개의 버전을 지원합니다.

객체 버전 관리는 StorageGRID 정보 라이프사이클 관리(ILM) 또는 S3 버킷 수명 주기 구성과 결합할 수 있습니다. 각 버킷에 대해 버전 관리를 명시적으로 활성화해야 합니다. 버킷에 버전 관리가 활성화되면 해당 버킷에 추가되는 모든 객체에 StorageGRID 시스템에서 생성된 버전 ID가 할당됩니다.

다단계 인증(MFA)을 사용하는 경우 삭제는 지원되지 않습니다.

> ⓘ 버전 관리는 StorageGRID 버전 10.3 이상으로 생성된 버킷에서만 활성화할 수 있습니다.

ILM 및 버전 관리

ILM 정책은 객체의 각 버전에 적용됩니다. ILM 스캔 프로세스는 모든 객체를 지속적으로 스캔하고 현재 ILM 정책에 따라 재평가합니다. ILM 정책을 변경하면 이전에 수집된 모든 객체에 적용됩니다. 버전 관리가 활성화된 경우 이전에 수집된 버전에도 적용됩니다. ILM 스캔은 새로운 ILM 변경 사항을 이전에 수집된 객체에 적용합니다.

버전 관리가 활성화된 버킷의 S3 객체의 경우, 버전 관리 지원을 통해 "비현재 시간"을 참조 시간으로 사용하는 ILM 규칙을 생성할 수 있습니다("ILM 규칙 생성 마법사의 1단계"의 "이 규칙을 이전 객체 버전에만 적용하시겠습니까?"라는 질문에 *예*를 선택). 객체가 업데이트되면 이전 버전은 비현재 버전이 됩니다. "비현재 시간" 필터를 사용하면 이전 버전 객체의 스토리지 사용량을 줄이는 정책을 만들 수 있습니다.

멀티파트 업로드 작업을 사용하여 객체의 새 버전을 업로드할 때, 원본 버전의 비현재 시간은 멀티파트 업로드가 완료된 시간이 아니라 새 버전에 대한 멀티파트 업로드가 생성된 시간을 반영합니다. 드물지만 원본 버전의 비현재 시간이 현재 버전의 시간보다 몇 시간 또는 며칠 전일 수 있습니다.

관련 정보

- "S3 버전 관리 객체가 삭제되는 방법"
- "S3 버전 관리 객체에 대한 ILM 규칙 및 정책(예제 4)".

S3 REST API를 사용하여 StorageGRID S3 Object Lock을 구성합니다

StorageGRID 시스템에서 전역 S3 객체 잠금 설정이 활성화된 경우, S3 객체 잠금이 활성화된 버킷을 생성할 수 있습니다. 각 버킷에 대한 기본 보존 기간 또는 각 객체 버전에 대한 보존 기간 설정을 지정할 수 있습니다.

버킷에 S3 Object Lock을 활성화하는 방법

StorageGRID 시스템에 대해 전역 S3 객체 잠금 설정이 활성화된 경우, 각 버킷을 생성할 때 선택적으로 S3 Object Lock을 활성화할 수 있습니다.

S3 객체 잠금은 버킷 생성 시에만 활성화할 수 있는 영구 설정입니다. 버킷 생성 후에는 S3 객체 잠금을 추가하거나 비활성화할 수 없습니다.

버킷에 S3 Object Lock을 활성화하려면 다음 방법 중 하나를 사용하십시오.

- 테넌트 관리자를 사용하여 버킷을 생성합니다. "S3 버킷 생성"을 참조하십시오.
- CreateBucket 요청과 `x-amz-bucket-object-lock-enabled` 요청 헤더를 사용하여 버킷을 생성합니다. "버킷에 대한 작업"을 참조하십시오.

S3 객체 잠금에는 버킷 버전 관리가 필요하며, 이는 버킷 생성 시 자동으로 활성화됩니다. 버킷의 버전 관리를 일시 중지할 수 없습니다. "객체 버전 관리"을 참조하십시오.

버킷의 기본 보존 설정

버킷에 대해 S3 Object Lock이 활성화된 경우, 선택적으로 버킷에 대한 기본 보존을 활성화하고 기본 보존 모드 및 기본 보존 기간을 지정할 수 있습니다.

기본 보존 모드

- 규정 준수 모드에서:
  - 해당 객체는 지정된 보존 기간이 만료될 때까지 삭제할 수 없습니다.
  - 객체의 보존 기간은 늘릴 수는 있지만 줄일 수는 없습니다.
  - 해당 날짜에 도달할 때까지 객체의 보존 기한을 제거할 수 없습니다.
- 거버넌스 모드에서:
  - `s3:BypassGovernanceRetention` 권한이 있는 사용자는 `x-amz-bypass-governance-retention: true` 요청 헤더를 사용하여 보존 설정을 우회할 수 있습니다.
  - 이러한 사용자는 보존 기한이 도래하기 전에 개체 버전을 삭제할 수 있습니다.

◦ 이러한 사용자는 개체의 보존 기간을 늘리거나 줄이거나 없앨 수 있습니다.

기본 보존 기간

각 버킷에는 연 또는 일 단위로 지정된 기본 보존 기간이 있을 수 있습니다.

버킷의 기본 보존 기간을 설정하는 방법

버킷의 기본 보존 기간을 설정하려면 다음 방법 중 하나를 사용하십시오.

- 테넌트 관리자에서 버킷 설정을 관리합니다. "S3 버킷 생성" 및 "S3 객체 잠금 기본 보존 기간 업데이트"을 참조하십시오.
- 버킷에 대한 PutObjectLockConfiguration 요청을 실행하여 기본 모드와 기본 일수 또는 연수를 지정합니다.

PutObjectLockConfiguration

PutObjectLockConfiguration 요청을 사용하면 S3 객체 잠금이 활성화된 버킷의 기본 보존 모드와 기본 보존 기간을 설정하고 수정할 수 있습니다. 또한 이전에 구성된 기본 보존 설정을 제거할 수도 있습니다.

새로운 객체 버전이 버킷에 수집될 때, `x-amz-object-lock-mode` 및 `x-amz-object-lock-retain-until-date`이(가) 지정되지 않은 경우 기본 보존 모드가 적용됩니다. `x-amz-object-lock-retain-until-date`이(가) 지정되지 않으면 기본 보존 기간을 사용하여 보존 종료일을 계산합니다.

객체 버전이 수집된 후 기본 보존 기간이 수정되더라도 객체 버전의 보존 만료일은 그대로 유지되며 새 기본 보존 기간을 사용하여 재계산되지 않습니다.

이 작업을 완료하려면 `s3:PutBucketObjectLockConfiguration` 권한이 있거나 루트 계정이어야 합니다.

The `Content-MD5` 요청 헤더는 PUT 요청에 반드시 지정해야 합니다.

요청 예시

이 예제는 버킷에 대해 S3 Object Lock을 활성화하고 기본 보존 모드를 COMPLIANCE로, 기본 보존 기간을 6년으로 설정합니다.

```
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>
```

버킷의 기본 보존 기간을 확인하는 방법

버킷에 S3 객체 잠금이 활성화되어 있는지 여부와 기본 보존 모드 및 보존 기간을 확인하려면 다음 방법 중 하나를 사용하십시오.

- 테넌트 관리자에서 버킷을 확인합니다. "S3 버킷 보기"을 참조하십시오.
- GetObjectLockConfiguration 요청을 실행합니다.

GetObjectLockConfiguration

GetObjectLockConfiguration 요청을 사용하면 버킷에 대해 S3 객체 잠금이 활성화되어 있는지 확인할 수 있으며, 활성화되어 있는 경우 버킷에 대해 기본 보존 모드 및 보존 기간이 구성되어 있는지 확인할 수 있습니다.

새로운 객체 버전이 버킷에 수집될 때, `x-amz-object-lock-mode`이 지정되지 않으면 기본 보존 모드가 적용됩니다. `x-amz-object-lock-retain-until-date`이 지정되지 않은 경우 기본 보존 기간이 보존 만료일 계산에 사용됩니다.

이 작업을 완료하려면 `s3:GetBucketObjectLockConfiguration` 권한이 있거나 루트 계정이어야 합니다.

요청 예시

```
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
```

응답 예시

```
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>
```

객체에 대한 보존 설정을 지정하는 방법

S3 객체 잠금이 활성화된 버킷에는 S3 객체 잠금 보존 설정이 있는 객체와 없는 객체가 혼합되어 포함될 수 있습니다.

객체 수준의 보존 설정은 S3 REST API를 사용하여 지정합니다. 객체에 대한 보존 설정은 버킷의 기본 보존 설정을 재정의합니다.

각 객체에 대해 다음과 같은 설정을 지정할 수 있습니다.

- 보존 모드: COMPLIANCE 또는 GOVERNANCE 중 하나.
- 보존 기한: 개체 버전을 StorageGRID에서 얼마나 오래 보존해야 하는지를 지정하는 날짜입니다.
  - 규정 준수 모드에서 보존 기한이 미래 날짜로 설정된 경우, 해당 개체를 검색할 수는 있지만 수정하거나 삭제할 수는 없습니다. 보존 기한은 늘릴 수 있지만, 줄이거나 삭제할 수는 없습니다.

    - 거버넌스 모드에서 특별 권한을 가진 사용자는 보존 기간 설정을 무시할 수 있습니다. 보존 기간이 만료되기 전에 개체 버전을 삭제할 수 있습니다. 또한 보존 기간을 늘리거나 줄이거나 아예 제거할 수도 있습니다.

- 법적 보류: 객체 버전에 법적 보류를 적용하면 해당 객체가 즉시 잠깁니다. 예를 들어, 조사 또는 법적 분쟁과 관련된 객체에 법적 보류를 적용해야 할 수 있습니다. 법적 보류에는 만료일이 없으며 명시적으로 해제될 때까지 유지됩니다.

  객체에 대한 법적 보존 설정은 보존 모드 및 보존 만료일과 무관합니다. 객체 버전이 법적 보존 상태에 있는 경우 누구도 해당 버전을 삭제할 수 없습니다.

버킷에 객체 버전을 추가할 때 S3 Object Lock 설정을 지정하려면 "PutObject", "CopyObject", 또는 "CreateMultipartUpload" 요청을 실행하십시오.

다음을 사용할 수 있습니다.

- `x-amz-object-lock-mode`이는 COMPLIANCE 또는 GOVERNANCE일 수 있습니다(대소문자 구분).

  > 을(를) 지정하는 경우 `x-amz-object-lock-mode`, 도 지정해야 합니다 `x-amz-object-lock-retain-until-date`.

- `x-amz-object-lock-retain-until-date`
    - 보존 기한 값은 반드시 형식이어야 합니다 `2020-08-10T21:46:00Z`. 소수점 이하 초 단위까지 허용되지만, 소수점 이하 3자리까지만 보존됩니다(밀리초 정밀도). 다른 ISO 8601 형식은 허용되지 않습니다.
    - 보존 기한은 미래 날짜로 설정해야 합니다.
- `x-amz-object-lock-legal-hold`

  법적 보류(Legal Hold)가 켜짐(대소문자 구분)인 경우 해당 객체는 법적 보류 상태가 됩니다. 법적 보류가 꺼짐(Off)인 경우 법적 보류가 적용되지 않습니다. 그 외의 값은 400 Bad Request(InvalidArgument) 오류를 발생시킵니다.

이러한 요청 헤더를 사용하는 경우 다음 제한 사항에 유의하십시오.

- `Content-MD5` 요청 헤더는 PutObject 요청에 `x-amz-object-lock-*` 요청 헤더가 있는 경우 필수입니다. `Content-MD5`은 CopyObject 또는 CreateMultipartUpload에는 필요하지 않습니다.
- 버킷에 S3 객체 잠금이 활성화되어 있지 않고 `x-amz-object-lock-*`요청 헤더가 있는 경우 400 잘못된 요청(InvalidRequest) 오류가 반환됩니다.
- PutObject 요청은 AWS 동작과 일치하도록 `x-amz-storage-class: REDUCED_REDUNDANCY`을 사용할 수 있습니다. 그러나 S3 객체 잠금이 활성화된 버킷에 객체를 수집할 경우 StorageGRID는 항상 이중 커밋 수집을 수행합니다.
- 이후의 GET 또는 HeadObject 버전 응답에는 구성되어 있고 요청 발신자에게 올바른 `s3:Get*` 권한이 있는 경우 헤더 `x-amz-object-lock-mode`, `x-amz-object-lock-retain-until-date` 및 `x-amz-object-lock-legal-hold`이 포함됩니다.

개체에 허용되는 최소 및 최대 보존 기간을 제한하려면 `s3:object-lock-remaining-retention-days` 정책 조건 키를 사용할 수 있습니다.

객체의 보존 설정을 업데이트하는 방법

기존 개체 버전의 법적 보존 또는 보존 기간 설정을 업데이트해야 하는 경우 다음 개체 하위 리소스 작업을 수행할 수 있습니다.

- `PutObjectLegalHold`

  새로운 법적 보류 값이 ON으로 설정되면 해당 개체는 법적 보류 상태가 됩니다. 법적 보류 값이 OFF로 설정되면 법적 보류가 해제됩니다.

- `PutObjectRetention`
  - 모드 값은 COMPLIANCE 또는 GOVERNANCE일 수 있습니다(대소문자 구분).
  - 보존 기한 값은 반드시 형식이어야 합니다 `2020-08-10T21:46:00Z`. 소수점 이하 초 단위까지 허용되지만, 소수점 이하 3자리까지만 보존됩니다(밀리초 정밀도). 다른 ISO 8601 형식은 허용되지 않습니다.
  - 객체 버전에 기존 보존 날짜가 있는 경우 해당 날짜를 늘릴 수만 있습니다. 새 값은 미래 날짜여야 합니다.

거버넌스 모드 사용 방법

`s3:BypassGovernanceRetention`권한이 있는 사용자는 `GOVERNANCE` 모드를 사용하는 객체의 활성 보존 설정을 무시할 수 있습니다. 모든 `DELETE` 또는 `PutObjectRetention` 작업에는 `x-amz-bypass-governance-retention:true`요청 헤더가 포함되어야 합니다. 이러한 사용자는 다음과 같은 추가 작업을 수행할 수 있습니다.

- DeleteObject 또는 DeleteObjects 작업을 수행하여 보존 기간이 경과하기 전에 개체 버전을 삭제합니다.

  법적 보존 조치가 적용된 개체는 삭제할 수 없습니다. 법적 보존 조치를 해제해야 합니다.

- PutObjectRetention 작업을 수행하여 객체의 보존 기간이 만료되기 전에 객체 버전의 모드를 GOVERNANCE에서 COMPLIANCE로 변경합니다.

  규정 준수 모드에서 거버넌스 모드로 변경하는 것은 절대 허용되지 않습니다.

- PutObjectRetention 작업을 수행하여 객체 버전의 보존 기간을 늘리거나, 줄이거나, 제거합니다.

관련 정보

- "S3 Object Lock으로 객체 관리"
- "S3 객체 잠금을 사용하여 객체 보존"
- "Amazon Simple Storage Service 사용자 가이드: 객체 잠금"

StorageGRID의 S3 라이프사이클 구성에 대해 알아보십시오

StorageGRID 시스템에서 특정 객체가 삭제되는 시점을 제어하기 위해 S3 수명 주기 구성을 생성할 수 있습니다.

이 섹션의 간단한 예제는 S3 수명 주기 구성을 통해 특정 S3 버킷에서 특정 객체가 삭제(만료)되는 시점을 제어하는 방법을 보여줍니다. 이 섹션의 예제는 설명 목적으로만 제공됩니다. S3 수명 주기 구성 생성에 대한 자세한 내용은 "Amazon Simple Storage Service 사용자 가이드: 객체 수명 주기 관리"을 참조하십시오. StorageGRID는 만료

작업만 지원하며 전환 작업은 지원하지 않습니다.

라이프사이클 구성이란 무엇인가요?

수명 주기 구성은 특정 S3 버킷의 객체에 적용되는 규칙 집합입니다. 각 규칙은 어떤 객체가 영향을 받는지, 그리고 해당 객체가 언제 만료되는지(특정 날짜 또는 일정 기간 후)를 지정합니다.

StorageGRID는 라이프사이클 구성에서 최대 1,000개의 라이프사이클 규칙을 지원합니다. 각 규칙에는 다음 XML 요소가 포함될 수 있습니다.

- 만료: 지정된 날짜가 되거나 객체가 수집된 시점으로부터 지정된 일수가 경과하면 객체를 삭제합니다.
- NoncurrentVersionExpiration: 개체가 비활성 상태가 된 시점부터 지정된 일수가 경과하면 개체를 삭제합니다.
- 필터(접두사, 태그)
- 상태
- ID

각 객체는 S3 버킷 수명 주기 또는 ILM 정책의 보존 설정을 따릅니다. S3 버킷 수명 주기가 구성된 경우, 버킷 수명 주기 필터와 일치하는 객체에 대해서는 수명 주기 만료 작업이 ILM 정책을 재정의합니다. 버킷 수명 주기 필터와 일치하지 않는 객체는 ILM 정책의 보존 설정을 사용합니다. 객체가 버킷 수명 주기 필터와 일치하고 만료 작업이 명시적으로 지정되지 않은 경우, ILM 정책의 보존 설정은 사용되지 않으며 객체 버전은 영구적으로 보존되는 것으로 간주됩니다. "S3 버킷 수명 주기 및 ILM 정책에 대한 우선순위 예시"을 참조하십시오.

결과적으로, ILM 규칙의 배치 지침이 해당 객체에 여전히 적용되더라도 객체가 그리드에서 제거될 수 있습니다. 또는, 객체에 대한 ILM 배치 지침의 유효 기간이 만료된 후에도 객체가 그리드에 남아 있을 수 있습니다. 자세한 내용은 "ILM이 객체의 수명 주기 전반에 걸쳐 작동하는 방식"을 참조하십시오.

> 버킷 수명 주기 구성은 S3 Object Lock이 활성화된 버킷에서 사용할 수 있지만, 버킷 수명 주기 구성은 레거시 규정 준수 버킷에서는 지원되지 않습니다.

StorageGRID는 수명 주기 구성을 관리하기 위해 다음 버킷 작업의 사용을 지원합니다.

- DeleteBucketLifecycle
- GetBucketLifecycleConfiguration
- PutBucketLifecycleConfiguration

라이프사이클 구성 생성

라이프사이클 구성을 만드는 첫 번째 단계로, 하나 이상의 규칙이 포함된 JSON 파일을 만듭니다. 예를 들어, 이 JSON 파일에는 다음과 같이 세 가지 규칙이 포함되어 있습니다.

1. 규칙 1은 접두사 `category1`/와 일치하고 `key2`값이 `tag2`인 객체에만 적용됩니다. `Expiration`매개변수는 필터와 일치하는 객체가 2020년 8월 22일 자정에 만료됨을 지정합니다.
2. 규칙 2는 접두사 `category2`/와 일치하는 객체에만 적용됩니다. 해당 `Expiration` 매개변수는 필터와 일치하는 객체가 수집된 후 100일 후에 만료되도록 지정합니다.

> 일수를 지정하는 규칙은 객체가 수집된 날짜를 기준으로 합니다. 현재 날짜가 수집 날짜에 지정된 일수를 더한 날짜를 초과하면 수명 주기 구성이 적용되는 즉시 일부 객체가 버킷에서 제거될 수 있습니다.

3. 규칙 3은 접두사 `category3/`와 일치하는 객체에만 적용됩니다. `Expiration` 매개변수는 일치하는 객체의 비현재 버전이 비현재 버전이 된 후 50일 후에 만료되도록 지정합니다.

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

버킷에 수명 주기 구성 적용

라이프사이클 구성 파일을 생성한 후 PutBucketLifecycleConfiguration 요청을 실행하여 해당 파일을 버킷에 적용합니다.

이 요청은 예제 파일의 수명 주기 구성을 `testbucket`라는 이름의 버킷에 있는 객체에 적용합니다.

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

버킷에 수명 주기 구성이 성공적으로 적용되었는지 확인하려면 GetBucketLifecycleConfiguration 요청을 실행하십시오. 예를 들면 다음과 같습니다.

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

성공적인 응답에는 방금 적용한 라이프사이클 구성이 나열됩니다.

버킷 수명 주기 만료가 객체에 적용되는지 확인합니다.

PutObject, HeadObject 또는 GetObject 요청을 실행할 때 수명 주기 구성의 만료 규칙이 특정 개체에 적용되는지 여부를 확인할 수 있습니다. 규칙이 적용되는 경우 응답에는 `Expiration` 개체가 만료되는 시점과 일치하는 만료 규칙을 나타내는 매개변수가 포함됩니다.

> 버킷 수명 주기가 ILM을 재정의하므로 `expiry-date` 표시되는 날짜는 객체가 실제로 삭제되는 날짜입니다. 자세한 내용은 "오브젝트 보존 기간 결정 방법"을 참조하십시오.

예를 들어, 이 PutObject 요청은 2020년 6월 22일에 발행되었으며 `testbucket` 버킷에 객체를 배치합니다.

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

성공 응답은 해당 객체가 100일 후(2020년 10월 1일)에 만료되며 수명 주기 구성의 규칙 2와 일치함을 나타냅니다.

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

예를 들어, 이 HeadObject 요청은 testbucket 버킷에 있는 동일한 객체의 메타데이터를 가져오는 데 사용되었습니다.

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

성공 응답에는 객체의 메타데이터가 포함되며 해당 객체가 100일 후에 만료되고 규칙 2를 충족했음을 나타냅니다.

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

버전 관리가 활성화된 버킷의 경우 `x-amz-expiration` 응답 헤더는 객체의 현재 버전에만 적용됩니다.

StorageGRID로 S3 REST API 구현 권장 사항

StorageGRID와 함께 사용하기 위해 S3 REST API를 구현할 때 다음 권장 사항을 따라야 합니다.

존재하지 않는 객체에 대한 HEAD 권장 사항

애플리케이션에서 객체가 실제로 존재하지 않을 것으로 예상되는 경로에 객체가 있는지 정기적으로 확인하는 경우 "Available" "일관성"을 사용해야 합니다. 예를 들어, 애플리케이션에서 PUT 작업을 수행하기 전에 해당 위치에 HEAD 요청을 보내는 경우 "Available" 일관성을 사용해야 합니다.

그렇지 않으면 HEAD 작업에서 객체를 찾지 못할 경우, 동일 사이트의 두 개 이상의 스토리지 노드를 사용할 수 없거나 원격 사이트에 연결할 수 없는 경우 500 내부 서버 오류가 많이 발생할 수 있습니다.

"PUT 버킷 일관성" 요청을 사용하여 각 버킷에 대한 "사용 가능" 일관성을 설정하거나 개별 API 작업의 요청 헤더에 일관성을 지정할 수 있습니다.

객체 키에 대한 권장 사항

버킷이 처음 생성된 시점을 기준으로 객체 키 이름 지정에 대한 다음 권장 사항을 따르십시오.

StorageGRID 11.4 이하 버전에서 생성된 버킷

- 객체 키의 처음 네 문자에 임의의 값을 사용하지 마십시오. 이는 이전 AWS의 키 접두사 권장 사항과 상반됩니다. 대신, `image`과 같이 임의적이지 않고 고유하지 않은 접두사를 사용하십시오.
- AWS의 이전 권장 사항에 따라 키 접두사에 임의적이고 고유한 문자를 사용하려면 객체 키 앞에 디렉터리 이름을 접두사로 붙이십시오. 즉, 다음 형식을 사용하십시오.

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

이 형식 대신에:

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

StorageGRID 11.4 이상에서 생성된 버킷

성능 최적화를 위해 객체 키 이름을 제한할 필요는 없습니다. 대부분의 경우 객체 키 이름의 처음 네 문자에 임의의 값을 사용할 수 있습니다.

> 단, S3 워크로드가 짧은 시간 후 모든 객체를 지속적으로 삭제하는 경우는 예외입니다. 이러한 사용 사례에서 성능 영향을 최소화하려면 수천 개의 객체마다 키 이름의 앞부분을 날짜와 같은 것으로 변경하는 것이 좋습니다. 예를 들어, S3 클라이언트가 일반적으로 초당 2,000개의 객체를 기록하고 ILM 또는 버킷 수명 주기 정책에 따라 3일 후에 모든 객체가 삭제된다고 가정해 보겠습니다. 성능 영향을 최소화하려면 다음과 같은 패턴을 사용하여 키 이름을 지정할 수 있습니다.
> `/mybucket/mydir/yyyymmddhhmmss-random_UUID.jpg`

"범위 읽기**"**에 대한 권장 사항

이 "저장된 객체를 압축하는 전역 옵션"옵션이 활성화된 경우, S3 클라이언트 애플리케이션은 반환할 바이트 범위를 지정하는 GetObject 작업을 수행하지 않아야 합니다. 이러한 "범위 읽기" 작업은 StorageGRID가 요청된 바이트에 액세스하기 위해 객체를 사실상 압축 해제해야 하므로 비효율적입니다. GetObject 작업 중 매우 큰 객체에서 작은 바이트 범위를 요청하는 경우는 특히 비효율적입니다. 예를 들어, 50GB로 압축된 객체에서 10MB 범위를 읽는 것은 비효율적입니다.

압축된 객체에서 범위를 읽어오는 경우 클라이언트 요청 시간이 초과될 수 있습니다.

> 객체를 압축해야 하고 클라이언트 애플리케이션에서 범위 읽기를 사용해야 하는 경우 애플리케이션의 읽기 시간 제한을 늘리십시오.

Amazon S3 REST API 지원

StorageGRID에서 지원되는 S3 REST API 작업 및 제한 사항

StorageGRID 시스템은 대부분의 작업을 지원하지만 몇 가지 제한 사항이 있는 Simple Storage Service API(API 버전 2006-03-01)를 구현합니다. S3 REST API 클라이언트 애플리케이션을 통합할 때는 구현 세부 사항을 이해해야 합니다.

StorageGRID 시스템은 가상 호스트 방식 요청과 경로 방식 요청을 모두 지원합니다.

날짜 처리

StorageGRID에서 구현한 S3 REST API는 유효한 HTTP 날짜 형식만 지원합니다.

StorageGRID 시스템은 날짜 값을 허용하는 헤더에 대해 유효한 HTTP 날짜 형식만 지원합니다. 날짜의 시간 부분은 그리니치 표준시(GMT) 형식 또는 시간대 오프셋(+0000을 지정해야 함)이 없는 협정 세계시(UTC) 형식으로 지정할 수 있습니다. 요청에 `x-amz-date` 헤더를 포함하면 Date 요청 헤더에 지정된 값을 재정의합니다. AWS 서명 버전 4를 사용하는 경우 날짜 헤더가 지원되지 않으므로 `x-amz-date` 헤더가 서명된 요청에 반드시 포함되어야 합니다.

일반적인 요청 헤더

StorageGRID 시스템은 "Amazon Simple Storage Service API 참조: 일반적인 요청 헤더"에서 정의한 일반적인 요청 헤더를 지원하지만, 한 가지 예외가 있습니다.

요청 헤더	구현
권한 부여	AWS Signature Version 2에 대한 완벽한 지원 다음과 같은 예외 사항을 제외하고 AWS Signature Version 4를 지원합니다. • 실제 페이로드 체크섬 값을 `x-amz-content-sha256`에 제공하면, 헤더에 `UNSIGNED-PAYLOAD` 값이 제공된 것처럼 검증 없이 값이 수락됩니다. `aws-chunked` 스트리밍을 의미하는 `x-amz-content-sha256` 헤더 값(예: STREAMING-AWS4-HMAC-SHA256-PAYLOAD)을 제공하면 청크 서명이 청크 데이터에 대해 검증되지 않습니다.

일반적인 응답 헤더

StorageGRID 시스템은 _Simple Storage Service API Reference_에 정의된 일반적인 응답 헤더를 모두 지원하지만, 한 가지 예외가 있습니다.

응답 헤더	구현
x-amz-id-2	사용하지 않음

StorageGRID S3 REST API가 요청을 인증하는 방법

StorageGRID 시스템은 S3 API를 사용하여 오브젝트에 대한 인증된 액세스와 익명 액세스를 모두 지원합니다.

S3 API는 S3 API 요청 인증을 위해 서명 버전 2와 서명 버전 4를 지원합니다.

인증된 요청에는 액세스 키 ID와 비밀 액세스 키를 사용하여 서명해야 합니다.

StorageGRID 시스템은 HTTP `Authorization` 헤더와 쿼리 매개변수를 사용하는 두 가지 인증 방법을 지원합니다.

HTTP Authorization 헤더 사용

HTTP `Authorization` 헤더는 버킷 정책에서 허용하는 익명 요청을 제외한 모든 S3 API 작업에서 사용됩니다. 이 `Authorization` 헤더에는 요청을 인증하는 데 필요한 모든 서명 정보가 포함되어 있습니다.

쿼리 매개변수 사용

쿼리 매개변수를 사용하여 URL에 인증 정보를 추가할 수 있습니다. 이를 URL 사전 서명이라고 하며, 특정 리소스에 대한 임시 액세스 권한을 부여하는 데 사용할 수 있습니다. 사전 서명된 URL을 사용하는 사용자는 리소스에 액세스하기 위해 비밀 액세스 키를 알 필요가 없으므로 제3자에게 리소스에 대한 제한된 액세스 권한을 제공할 수 있습니다.

StorageGRID에서 지원되는 서비스 작업

StorageGRID 시스템은 서비스에 대해 다음과 같은 작업을 지원합니다.

작업	구현
ListBuckets (이전 명칭: GET 서비스)	모든 Amazon S3 REST API 동작에 맞춰 구현되었습니다. 예고 없이 변경될 수 있습니다.
저장 공간 사용량 가져오기	StorageGRID "저장 공간 사용량 가져오기" 요청은 계정에서 사용 중인 총 스토리지 용량과 해당 계정에 연결된 각 버킷의 스토리지 용량을 알려줍니다. 이는 경로가 /이고 사용자 지정 쿼리 매개변수 (`?x-ntap-sg-usage`가 추가된 서비스에 대한 작업입니다.
OPTIONS /	클라이언트 애플리케이션은 S3 인증 자격 증명을 제공하지 않고 스토리지 노드의 S3 포트에 `OPTIONS /` 요청을 발행하여 스토리지 노드의 가용 여부를 확인할 수 있습니다. 이 요청은 모니터링에 사용하거나 외부 로드 밸런서가 스토리지 노드의 다운 여부를 식별하는 데 사용할 수 있습니다.

StorageGRID의 S3 버킷 작업 및 구현 세부 정보

StorageGRID 시스템은 각 S3 테넌트 계정당 최대 5,000개의 버킷을 지원합니다.

각 그리드는 최대 100,000개의 버킷을 가질 수 있습니다.

5,000개의 버킷을 지원하려면 그리드의 각 스토리지 노드에 최소 64GB의 RAM이 있어야 합니다.

버킷 이름 제한은 AWS 미국 표준 지역 제한을 따르지만, S3 가상 호스팅 스타일 요청을 지원하려면 DNS 명명 규칙에 따라 추가로 제한해야 합니다.

자세한 내용은 다음을 참조하십시오.

- "Amazon Simple Storage Service 사용자 가이드: 버킷 할당량, 제한 사항 및 제약 조건"
- "S3 엔드포인트 도메인 이름 구성"

ListObjects(버킷 가져오기) 및 ListObjectVersions(버킷 객체 버전 가져오기) 작업은 StorageGRID "일관성 값"를 지원합니다.

개별 버킷에 대해 마지막 액세스 시간 업데이트가 활성화되었는지 또는 비활성화되었는지 확인할 수 있습니다. "GET 버킷 마지막 액세스 시간"을 참조하십시오.

다음 표는 StorageGRID S3 REST API 버킷 작업을 구현하는 방식을 설명합니다. 이러한 작업을 수행하려면 계정에 필요한 액세스 자격 증명을 제공해야 합니다.

작업	구현
CreateBucket	새 버킷을 생성합니다. 버킷을 생성함으로써 해당 버킷의 소유자가 됩니다. • 버킷 이름은 다음 규칙을 준수해야 합니다. ◦ 각 StorageGRID 시스템 전체에서 고유해야 합니다(테넌트 계정 내에서만 고유한 것이 아님). ◦ DNS 규정을 준수해야 합니다. ◦ 최소 3자, 최대 63자여야 합니다. ◦ 하나 이상의 레이블로 구성될 수 있으며, 인접한 레이블 간에는 마침표가 있어야 합니다. 각 레이블은 소문자 또는 숫자로 시작하고 끝나야 하며, 소문자, 숫자 및 하이픈만 사용할 수 있습니다. ◦ 텍스트 형식의 IP 주소처럼 보이지 않아야 합니다. ◦ 가상 호스팅 스타일 요청에는 마침표를 사용해서는 안 됩니다. 마침표는 서버 와일드카드 인증서 검증에 문제를 일으킬 수 있습니다. • 기본적으로 버킷은 `us-east-1` 리전에 생성됩니다. 하지만 요청 본문의 `LocationConstraint` 요청 요소를 사용하여 다른 리전을 지정할 수 있습니다. `LocationConstraint` 요소를 사용할 때는 Grid Manager 또는 Grid Management API를 사용하여 정의된 리전의 정확한 이름을 지정해야 합니다. 사용할 리전 이름을 모르는 경우 시스템 관리자에게 문의하십시오. 참고: CreateBucket 요청이 StorageGRID에 정의되지 않은 지역을 사용하는 경우 오류가 발생합니다. • `x-amz-bucket-object-lock-enabled` 요청 헤더를 포함하여 S3 Object Lock이 활성화된 버킷을 생성할 수 있습니다. "S3 REST API를 사용하여 S3 Object Lock 구성"을 참조하십시오. 버킷을 생성할 때 S3 객체 잠금을 활성화해야 합니다. 버킷 생성 후에는 S3 객체 잠금을 추가하거나 비활성화할 수 없습니다. S3 객체 잠금은 버킷 버전 관리가 필요하며, 이는 버킷 생성 시 자동으로 활성화됩니다.
DeleteBucket	버킷을 삭제합니다.
DeleteBucketCors	버킷의 CORS 구성을 삭제합니다.
DeleteBucketEncryption	버킷에서 기본 암호화를 삭제합니다. 기존에 암호화된 객체는 암호화된 상태로 유지되지만, 버킷에 새로 추가되는 객체는 암호화되지 않습니다.
DeleteBucketLifecycle	버킷에서 라이프사이클 구성을 삭제합니다. "S3 라이프사이클 구성 생성"을 참조하십시오.
DeleteBucketPolicy	버킷에 연결된 정책을 삭제합니다.
DeleteBucketReplication	버킷에 연결된 복제 구성을 삭제합니다.

작업	구현
DeleteBucketTagging	`tagging` 하위 리소스를 사용하여 버킷에서 모든 태그를 제거합니다. 주의: 이 버킷에 기본값이 아닌 ILM 정책 태그가 설정되어 있는 경우, `NTAP-SG-ILM-BUCKET-TAG` 버킷 태그와 할당된 값이 존재합니다. `NTAP-SG-ILM-BUCKET-TAG` 버킷 태그가 있는 경우 DeleteBucketTagging 요청을 실행하지 마십시오. 대신, `NTAP-SG-ILM-BUCKET-TAG` 태그와 할당된 값만 포함하여 PutBucketTagging 요청을 실행하면 버킷에서 다른 모든 태그가 제거됩니다. `NTAP-SG-ILM-BUCKET-TAG` 버킷 태그를 수정하거나 삭제하지 마십시오.
GetBucketAcl	버킷 소유자에게 버킷에 대한 모든 액세스 권한이 있음을 나타내는 긍정적인 응답과 버킷 소유자의 ID, DisplayName 및 Permission을 반환합니다.
GetBucketCors	버킷의 `cors` 구성을 반환합니다.
GetBucketEncryption	버킷의 기본 암호화 구성을 반환합니다.
GetBucketLifecycleConfiguration (이전 명칭: GET Bucket lifecycle)	버킷의 수명 주기 구성을 반환합니다. "S3 라이프사이클 구성 생성"을 참조하세요.
GetBucketLocation	`LocationConstraint` 요소를 사용하여 `CreateBucket` 요청에서 설정된 지역을 반환합니다. 버킷의 지역이 `us-east-1`인 경우, 지역에 대해 빈 문자열이 반환됩니다.
GetBucketNotificationConfiguration (이전 명칭: GET Bucket notification)	버킷에 연결된 알림 구성을 반환합니다.
GetBucketPolicy	버킷에 연결된 정책을 반환합니다.
GetBucketReplication	버킷에 연결된 복제 구성을 반환합니다.
GetBucketTagging	`tagging` 하위 리소스를 사용하여 버킷의 모든 태그를 반환합니다. 주의: 이 버킷에 기본값이 아닌 ILM 정책 태그가 설정된 경우, `NTAP-SG-ILM-BUCKET-TAG` 버킷 태그에 값이 할당됩니다. 이 태그를 수정하거나 삭제하지 마십시오.

작업	구현
GetBucketVersioning	이 구현에서는 `versioning` 하위 리소스를 사용하여 버킷의 버전 관리 상태를 반환합니다. • *blank*: 버전 관리가 활성화된 적이 없습니다(버킷이 "Unversioned"입니다). • 활성화됨: 버전 관리가 활성화되었습니다. • 일시 중단됨: 이전에 버전 관리가 활성화되었으나 현재 일시 중단되었습니다.
GetObjectLockConfiguration	구성된 경우 버킷의 기본 보존 모드와 기본 보존 기간을 반환합니다. "S3 REST API를 사용하여 S3 Object Lock 구성"을 참조하십시오.
HeadBucket	버킷이 존재하는지, 그리고 해당 버킷에 액세스할 권한이 있는지 확인합니다. 이 연산의 결과는 다음과 같습니다. • `x-ntap-sg-bucket-id`: 버킷의 UUID(UUID 형식)입니다. • `x-ntap-sg-trace-id`: 해당 요청의 고유 추적 ID입니다.
ListObjects 및 ListObjectsV2 (이전 명칭: GET Bucket)	버킷에 있는 객체 중 일부 또는 전부(최대 1,000개)를 반환합니다. 객체의 스토리지 클래스는 해당 객체가 `REDUCED_REDUNDANCY` 스토리지 클래스 옵션을 사용하여 수집된 경우에도 두 가지 값 중 하나를 가질 수 있습니다. • `STANDARD`, 이는 해당 객체가 스토리지 노드로 구성된 스토리지 풀에 저장되어 있음을 나타냅니다. • `GLACIER`이는 해당 객체가 클라우드 스토리지 풀에서 지정한 외부 버킷으로 이동되었음을 나타냅니다. 버킷에 동일한 접두사를 가진 삭제된 키가 대량으로 포함되어 있는 경우, 응답에 키가 포함되지 않은 `CommonPrefixes`이 일부 포함될 수 있습니다. HeadObject 및 ListObject 요청의 경우 StorageGRID는 LastModified 타임스탬프를 서로 다른 정밀도로 반환하는 반면 AWS는 다음 예에서와 같이 동일한 정밀도로 타임스탬프를 반환합니다. • 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 (이전 명칭: GET Bucket Object versions)	버킷에 대한 읽기 권한이 있는 경우, `versions` 하위 리소스와 함께 이 작업을 사용하면 버킷에 있는 모든 객체 버전의 메타데이터가 나열됩니다.

작업	구현
PutBucketCors	버킷이 교차 출처 요청을 처리할 수 있도록 버킷의 CORS 구성을 설정합니다. 교차 출처 리소스 공유(CORS)는 한 도메인의 클라이언트 웹 애플리케이션이 다른 도메인의 리소스에 액세스할 수 있도록 하는 보안 메커니즘입니다. 예를 들어, 그래픽을 저장하는 데 `images`라는 S3 버킷을 사용한다고 가정해 보겠습니다. `images` 버킷의 CORS 구성을 설정하면 해당 버킷의 이미지를 `http://www.example.com` 웹사이트에 표시할 수 있습니다.
PutBucketEncryption	기존 버킷의 기본 암호화 상태를 설정합니다. 버킷 수준 암호화가 활성화되면 버킷에 새로 추가되는 모든 객체가 암호화됩니다. StorageGRID는 StorageGRID에서 관리하는 키를 사용한 서버 측 암호화를 지원합니다. 서버 측 암호화 구성 규칙을 지정할 때 `SSEAlgorithm`매개변수를 `AES256`로 설정하고, `KMSMasterKeyID`매개변수는 사용하지 마십시오. 버킷의 기본 암호화 구성은 객체 업로드 요청에 이미 암호화가 지정된 경우(즉, 요청에 `x-amz-server-side-encryption-*` 요청 헤더가 포함된 경우) 무시됩니다.
PutBucketLifecycleConfiguration (이전 명칭: PUT Bucket lifecycle)	버킷에 대한 새 수명 주기 구성을 생성하거나 기존 수명 주기 구성을 대체합니다. StorageGRID는 수명 주기 구성에서 최대 1,000개의 수명 주기 규칙을 지원합니다. 각 규칙에는 다음 XML 요소가 포함될 수 있습니다. • 만료(일, 날짜, ExpiredObjectDeleteMarker) • NoncurrentVersionExpiration (NewerNoncurrentVersions, NoncurrentDays) • 필터(접두사, 태그) • 상태 • ID StorageGRID는 다음 작업을 지원하지 않습니다. • AbortIncompleteMultipartUpload • 전환 을 참조하십시오. "S3 라이프사이클 구성 생성". 버킷 수명 주기의 만료 작업이 ILM 배치 지침과 어떻게 상호 작용하는지 이해하려면 "ILM이 객체의 수명 주기 전반에 걸쳐 작동하는 방식"을 참조하십시오. 참고: 버킷 수명 주기 구성은 S3 Object Lock이 활성화된 버킷에서 사용할 수 있지만, 버킷 수명 주기 구성은 레거시 규정 준수 버킷에서는 지원되지 않습니다.

작업	구현

PutBucketNotificationConfiguration
 	
(이전 명칭: PUT Bucket notification)
	요청 본문에 포함된 알림 구성 XML을 사용하여 버킷에 대한 알림을 구성합니다. 다음 구현 세부 사항에 유의해야 합니다.
	
	• StorageGRID는 Amazon Simple Notification Service(Amazon SNS) 토픽, Kafka 토픽 또는 웹훅 엔드포인트를 대상으로 지원합니다. Simple Queue Service(SQS) 또는 AWS Lambda 엔드포인트는 지원하지 않습니다.
	• 알림 대상은 StorageGRID 엔드포인트의 URN으로 지정해야 합니다. 엔드포인트는 테넌트 관리자 또는 테넌트 관리 API를 사용하여 생성할 수 있습니다.
	
	알림 구성이 성공하려면 엔드포인트가 존재해야 합니다. 엔드포인트가 존재하지 않으면 `400 Bad Request` 오류가 코드 `InvalidArgument`와 함께 반환됩니다.
	
	• 다음 이벤트 유형에 대해서는 알림을 설정할 수 없습니다. 이러한 이벤트 유형은 지원되지 않습니다.
	◦ `s3:ReducedRedundancyLostObject`
	◦ `s3:ObjectRestore:Completed`
	• StorageGRID에서 전송된 이벤트 알림은 표준 JSON 형식을 사용하지만, 다음 목록에 표시된 것처럼 일부 키는 포함하지 않고 다른 키에는 특정 값을 사용합니다.
	◦ **eventSource**
	
	`sgws:s3`
	
	◦ **awsRegion**
	
	포함되지 않음
	
	◦ **x-amz-id-2**
	
	포함되지 않음
	
	◦ **arn**
	
	`urn:sgws:s3:::bucket_name`

PutBucketPolicy	버킷에 연결된 정책을 설정합니다. 자세한 내용은 "버킷 및 그룹 액세스 정책 사용"을 참조하십시오.

작업	구현
PutBucketReplication	요청 본문에 제공된 복제 구성 XML을 사용하여 버킷에 대해 "StorageGRID CloudMirror 복제"을(를) 구성합니다. CloudMirror 복제의 경우 다음 구현 세부 정보를 숙지해야 합니다. • StorageGRID는 복제 구성의 V1만 지원합니다. 즉, StorageGRID는 규칙에 대한 `Filter`요소 사용을 지원하지 않으며 객체 버전 삭제에 대한 V1 규칙을 따릅니다. 자세한 내용은 "Amazon Simple Storage Service 사용자 가이드: 복제 구성"을 참조하십시오. • 버킷 복제는 버전이 지정된 버킷 또는 버전이 지정되지 않은 버킷 모두에서 구성할 수 있습니다. • 복제 구성 XML의 각 규칙에서 서로 다른 대상 버킷을 지정할 수 있습니다. 소스 버킷은 둘 이상의 대상 버킷으로 복제할 수 있습니다. • 대상 버킷은 테넌트 관리자 또는 테넌트 관리 API에 지정된 StorageGRID 엔드포인트의 URN으로 지정해야 합니다. "CloudMirror 복제 구성"을 참조하십시오. 복제 구성이 성공하려면 엔드포인트가 존재해야 합니다. 엔드포인트가 존재하지 않으면 요청이 `400 Bad Request`로 실패합니다. 오류 메시지는 다음과 같습니다. `Unable to save the replication policy. The specified endpoint URN does not exist: URN.` • 구성 XML에서 `Role`을(를) 지정할 필요가 없습니다. 이 값은 StorageGRID에서 사용되지 않으며 제출하더라도 무시됩니다. • 구성 XML에서 스토리지 클래스를 생략하면 StorageGRID는 기본적으로 `STANDARD` 스토리지 클래스를 사용합니다. • 원본 버킷에서 객체를 삭제하거나 원본 버킷 자체를 삭제하는 경우, 리전 간 복제 동작은 다음과 같습니다. ◦ 객체 또는 버킷이 복제되기 전에 삭제하면 해당 객체/버킷은 복제되지 않으며 알림도 표시되지 않습니다. ◦ 객체 또는 버킷이 복제된 후 삭제하면 StorageGRID는 리전 간 복제 V1에 대한 표준 Amazon S3 삭제 동작을 따릅니다.

작업	구현
PutBucketTagging	이 `tagging` 하위 리소스를 사용하여 버킷에 대한 태그 세트를 추가하거나 업데이트합니다. 버킷 태그를 추가할 때 다음 제한 사항에 유의하십시오. • StorageGRID 와 Amazon S3 모두 버킷당 최대 50개의 태그를 지원합니다. • 버킷과 연결된 태그는 고유한 태그 키를 가져야 합니다. 태그 키는 최대 128자의 유니코드 문자로 구성될 수 있습니다. • 태그 값은 최대 256자의 유니코드 문자까지 사용할 수 있습니다. • 키와 값은 대소문자를 구분합니다. 주의: 이 버킷에 기본값이 아닌 ILM 정책 태그가 설정된 경우, `NTAP-SG-ILM-BUCKET-TAG` 해당 버킷 태그에 값이 할당됩니다. 모든 PutBucketTagging 요청에 `NTAP-SG-ILM-BUCKET-TAG` 버킷 태그와 할당된 값이 포함되어야 합니다. 이 태그를 수정하거나 삭제하지 마십시오. 참고: 이 작업은 버킷에 이미 있는 태그를 덮어씁니다. 기존 태그 중 누락된 태그가 있는 경우 해당 태그는 버킷에서 제거됩니다.
PutBucketVersioning	`versioning` 하위 리소스를 사용하여 기존 버킷의 버전 관리 상태를 설정합니다. 다음 값 중 하나를 사용하여 버전 관리 상태를 설정할 수 있습니다. • 활성화됨: 버킷의 객체에 대한 버전 관리를 활성화합니다. 버킷에 추가되는 모든 객체는 고유한 버전 ID를 받습니다. • 일시 중단됨: 버킷의 객체에 대한 버전 관리를 비활성화합니다. 버킷에 추가되는 모든 객체는 버전 ID `null`를 받게 됩니다.
PutObjectLockConfiguration	버킷의 기본 보존 모드 및 기본 보존 기간을 구성하거나 제거합니다. 기본 보존 기간이 수정되더라도 기존 객체 버전의 보존 만료일은 그대로 유지되며 새 기본 보존 기간을 사용하여 재계산되지 않습니다. 자세한 내용은 "S3 REST API를 사용하여 S3 Object Lock 구성"을 참조하십시오.

객체에 대한 연산

StorageGRID에서 객체에 대한 **S3 REST API** 작업을 구현하는 방법

이 섹션에서는 StorageGRID 시스템이 객체에 대한 S3 REST API 작업을 구현하는 방법을 설명합니다.

다음 조건은 모든 객체 작업에 적용됩니다.

- StorageGRID "일관성 값"은(는) 다음을 제외한 모든 객체 작업에서 지원됩니다.

- GetObjectAcl
- `OPTIONS /`
- PutObjectLegalHold
- PutObjectRetention
- SelectObjectContent

- 두 클라이언트가 동일한 키에 쓰기를 시도하는 등 클라이언트 요청이 충돌하는 경우 "최신 요청 우선" 원칙에 따라 처리됩니다. "최신 요청 우선" 평가 시점은 S3 클라이언트가 작업을 시작한 시점이 아니라 StorageGRID 시스템이 해당 요청을 완료한 시점을 기준으로 합니다.
- StorageGRID 버킷의 모든 객체는 버킷 소유자의 소유이며, 여기에는 익명 사용자 또는 다른 계정이 생성한 객체도 포함됩니다.

다음 표는 StorageGRID S3 REST API 객체 작업을 구현하는 방식을 설명합니다.

작업	구현
DeleteObject	DeleteObject 요청을 처리할 때 StorageGRID는 모든 저장 위치에서 해당 객체의 모든 복사본을 즉시 제거하려고 시도합니다. 제거에 성공하면 StorageGRID는 클라이언트에 즉시 응답을 반환합니다. 모든 복사본을 30초 이내에 제거할 수 없는 경우(예: 특정 위치를 일시적으로 사용할 수 없는 경우) StorageGRID는 제거를 위해 복사본을 대기열에 추가한 후 클라이언트에 성공을 알립니다. 제한 사항 • 다중 요소 인증(MFA) 및 응답 헤더 `x-amz-mfa`는 지원되지 않습니다. • `If-None` 및 `If-None-Match` 헤더는 허용되지만 기능하지 않습니다. 버전 관리 특정 버전을 삭제하려면 요청자가 버킷 소유자여야 하며 `versionId` 하위 리소스를 사용해야 합니다. 이 하위 리소스를 사용하면 해당 버전이 영구적으로 삭제됩니다. `versionId`가 삭제 마커에 해당하는 경우 응답 헤더 `x-amz-delete-marker`가 `true`로 설정되어 반환됩니다. • 버전 관리가 활성화된 버킷에서 `versionId` 하위 리소스 없이 객체를 삭제하면 삭제 마커가 생성됩니다. 삭제 마커에 대한 `versionId`는 `x-amz-version-id` 응답 헤더를 사용하여 반환되며, `x-amz-delete-marker` 응답 헤더는 `true`로 설정되어 반환됩니다. • 버전 관리가 일시 중단된 버킷에서 `versionId` 하위 리소스 없이 객체를 삭제하면 이미 존재하는 'null' 버전 또는 'null' 삭제 마커가 영구적으로 삭제되고 새로운 'null' 삭제 마커가 생성됩니다. `x-amz-delete-marker` 응답 헤더는 `true`으로 설정되어 반환됩니다. 참고: 경우에 따라 하나의 객체에 여러 개의 삭제 표시가 존재할 수 있습니다. "S3 REST API를 사용하여 S3 Object Lock 구성"을 참조하여 GOVERNANCE 모드에서 객체 버전을 삭제하는 방법을 알아보십시오.

작업	구현
DeleteObjects (이전 명칭: DELETE Multiple Objects)	다중 요소 인증(MFA) 및 응답 헤더 `x-amz-mfa`는 지원되지 않습니다. 하나의 요청 메시지에서 여러 객체를 삭제할 수 있습니다. "S3 REST API를 사용하여 S3 Object Lock 구성"을 참조하여 GOVERNANCE 모드에서 객체 버전을 삭제하는 방법을 알아보십시오.
DeleteObjectTagging	`tagging` 하위 리소스를 사용하여 객체에서 모든 태그를 제거합니다. 버전 관리 요청에 `versionId` 쿼리 매개변수가 지정되지 않은 경우, 해당 작업은 버전 관리 버킷에서 객체의 최신 버전에 있는 모든 태그를 삭제합니다. 현재 객체 버전이 삭제 마커인 경우, "MethodNotAllowed" 상태가 `x-amz-delete-marker` 응답 헤더가 `true`로 설정된 상태로 반환됩니다.
GetObject	"GetObject"
GetObjectAcl	계정에 필요한 액세스 자격 증명이 제공되면 작업은 긍정적인 응답과 함께 개체 소유자의 ID, DisplayName 및 권한을 반환하여 소유자가 해당 개체에 대한 모든 액세스 권한을 가지고 있음을 나타냅니다.
GetObjectLegalHold	"S3 REST API를 사용하여 S3 Object Lock 구성"
GetObjectRetention	"S3 REST API를 사용하여 S3 Object Lock 구성"
GetObjectTagging	`tagging` 하위 리소스를 사용하여 객체의 모든 태그를 반환합니다. 버전 관리 요청에 `versionId` 쿼리 매개변수가 지정되지 않은 경우, 해당 작업은 버전 관리 버킷에서 객체의 최신 버전의 모든 태그를 반환합니다. 현재 객체 버전이 삭제 마커인 경우, `x-amz-delete-marker` 응답 헤더가 `true`로 설정된 "MethodNotAllowed" 상태가 반환됩니다.
HeadObject	"HeadObject"
RestoreObject	"RestoreObject"
PutObject	"PutObject"

작업	구현
CopyObject (이전 명칭: PUT Object - Copy)	"CopyObject"
PutObjectLegalHold	"S3 REST API를 사용하여 S3 Object Lock 구성"
PutObjectRetention	"S3 REST API를 사용하여 S3 Object Lock 구성"
PutObjectTagging	`tagging` 하위 리소스를 사용하여 기존 개체에 태그 세트를 추가합니다. 객체 태그 제한 새 객체를 업로드할 때 태그를 추가하거나 기존 객체에 태그를 추가할 수 있습니다. StorageGRID와 Amazon S3 모두 객체당 최대 10개의 태그를 지원합니다. 객체와 연결된 태그는 고유한 태그 키를 가져야 합니다. 태그 키는 최대 128자의 유니코드 문자, 태그 값은 최대 256자의 유니코드 문자까지 사용할 수 있습니다. 키와 값은 대소문자를 구분합니다. 태그 업데이트 및 수집 동작 PutObjectTagging을 사용하여 객체의 태그를 업데이트하면 StorageGRID는 해당 객체를 다시 수집하지 않습니다. 즉, 일치하는 ILM 규칙에 지정된 수집 동작 옵션이 사용되지 않습니다. 업데이트로 인해 발생하는 객체 배치 변경 사항은 일반적인 백그라운드 ILM 프로세스에 의해 ILM이 재평가될 때 적용됩니다. 즉, ILM 규칙에서 수집 동작에 대해 엄격(Strict) 옵션을 사용하는 경우, 필요한 객체 배치가 불가능한 경우(예: 새로 필요한 위치를 사용할 수 없는 경우) 아무런 조치도 취해지지 않습니다. 업데이트된 객체는 필요한 배치가 가능해질 때까지 현재 위치를 유지합니다. 충돌 해결 두 클라이언트가 동일한 키에 쓰기를 시도하는 등 클라이언트 요청이 충돌하는 경우 "최신 요청 우선" 원칙에 따라 처리됩니다. "최신 요청 우선" 평가 시점은 S3 클라이언트가 작업을 시작한 시점이 아니라 StorageGRID 시스템이 해당 요청을 완료한 시점을 기준으로 합니다. 버전 관리 요청에 `versionId` 쿼리 매개변수가 지정되지 않은 경우, 해당 작업은 버전 관리 버킷에서 객체의 최신 버전에 태그를 추가합니다. 객체의 현재 버전이 삭제 마커인 경우, "MethodNotAllowed" 상태가 반환되며 `x-amz-delete-marker` 응답 헤더가 `true`로 설정됩니다.
SelectObjectContent	"SelectObjectContent"

StorageGRID에서 지원되는 Amazon S3 Select 작업

StorageGRID는 "SelectObjectContent 명령"에 대해 다음과 같은 Amazon S3 Select 절, 데이터 유형 및 연산자를 지원합니다.

> 목록에 없는 항목은 지원되지 않습니다.

구문에 대해서는 "SelectObjectContent"을 참조하십시오. S3 Select에 대한 자세한 내용은 "AWS S3 Select 설명서"을 참조하십시오.

S3 Select가 활성화된 테넌트 계정만 SelectObjectContent 쿼리를 실행할 수 있습니다. 자세한 내용은 "S3 Select 사용을 위한 고려 사항 및 요구 사항"을 참조하십시오.

조항

- SELECT 목록
- FROM 절
- WHERE 절
- LIMIT 절

데이터 유형

- 불
- 정수
- 문자열
- float
- 십진수, 숫자
- 타임스탬프

운영자

논리 연산자

- AND
- NOT
- 또는

비교 연산자

- <
- >
- ⇐
- >=
- =

- =
- <>
- !=
- 사이
- IN

패턴 매칭 연산자

- LIKE
- _
- %

단일 연산자

- IS NULL
- NULL이 아닙니다

수학 연산자

- +
- -
- *
- /
- %

StorageGRID는 Amazon S3 Select 연산자 우선순위를 따릅니다.

집계 함수

- AVG()
- COUNT(*)
- MAX()
- MIN()
- SUM()

조건 함수

- 사례
- COALESCE
- NULLIF

변환 함수

- CAST(지원되는 데이터 형식의 경우)

날짜 함수

- DATE_ADD
- DATE_DIFF
- 발췌
- TO_STRING
- TO_TIMESTAMP
- UTCNOW

문자열 함수

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

StorageGRID에서 서버 측 암호화를 사용하는 방법

서버 측 암호화를 사용하면 저장된 객체 데이터를 보호할 수 있습니다. StorageGRID는 객체에 데이터를 기록할 때 데이터를 암호화하고, 객체에 접근할 때 데이터를 복호화합니다.

${post_edited_translations.segment}

- **SSE(StorageGRID**에서 관리하는 키를 사용한 서버 측 암호화**)**: S3 요청을 통해 객체를 저장하면 StorageGRID가 고유한 키로 객체를 암호화합니다. S3 요청을 통해 객체를 검색하면 StorageGRID가 저장된 키를 사용하여 객체를 복호화합니다.
- **SSE-C(**고객 제공 키를 사용한 서버 측 암호화**)**: 객체를 저장하기 위해 S3 요청을 보낼 때 자체 암호화 키를 제공합니다. 객체를 검색할 때 요청의 일부로 동일한 암호화 키를 제공합니다. 두 암호화 키가 일치하면 객체가 복호화되고 객체 데이터가 반환됩니다.

  ${post_edited_translations.segment}

  사용자가 제공하는 암호화 키는 절대 저장되지 않습니다. 암호화 키를 분실하면 해당 객체도 함께 분실하게 됩니다.

  객체가 SSE 또는 SSE-C로 암호화된 경우 버킷 수준 또는 그리드 수준의 암호화 설정은 무시됩니다.

${post_edited_translations.segment}

${post_edited_translations.segment}

`x-amz-server-side-encryption`

SSE 요청 헤더는 다음 객체 작업에서 지원됩니다.

- "PutObject"
- "CopyObject"
- "CreateMultipartUpload"

SSE-C 사용

${post_edited_translations.segment}

요청 헤더	설명
`x-amz-server-side-encryption-customer-algorithm`	암호화 알고리즘을 지정합니다. 헤더 값은 `AES256`이어야 합니다.
`x-amz-server-side-encryption-customer-key`	오브젝트를 암호화하거나 복호화하는 데 사용할 암호화 키를 지정합니다. 키 값은 base64로 인코딩된 256비트여야 합니다.
`x-amz-server-side-encryption-customer-key-MD5`	RFC 1321에 따라 암호화 키가 오류 없이 전송되었는지 확인하는 데 사용되는 암호화 키의 MD5 다이제스트를 지정합니다. MD5 다이제스트 값은 base64로 인코딩된 128비트여야 합니다.

${post_edited_translations.segment}

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

${post_edited_translations.segment}

SSE-C를 사용하기 전에 다음 사항을 고려하십시오.

- https를 사용해야 합니다.

> StorageGRID는 SSE-C를 사용할 때 http를 통해 전송된 모든 요청을 거부합니다. 보안상 이유로, 실수로 http를 통해 전송한 모든 키는 유출된 것으로 간주해야 합니다. 해당 키를 폐기하고 적절히 교체하십시오.

- ${post_edited_translations.segment}
- 암호화 키와 오브젝트의 매핑을 관리해야 합니다. StorageGRID는 암호화 키를 저장하지 않습니다. 각 오브젝트에 대해 제공하는 암호화 키를 추적할 책임은 사용자에게 있습니다.
- 버킷의 버전 관리가 활성화된 경우, 각 오브젝트 버전에는 자체 암호화 키가 있어야 합니다. 각 오브젝트 버전에 사용된 암호화 키를 추적할 책임은 사용자에게 있습니다.

- 암호화 키를 클라이언트 측에서 관리하므로 키 순환과 같은 추가적인 보안 조치도 클라이언트 측에서 관리해야 합니다.

> 사용자가 제공하는 암호화 키는 절대 저장되지 않습니다. 암호화 키를 분실하면 해당 객체도 함께 분실하게 됩니다.

- 버킷에 대해 크로스 그리드 복제 또는 CloudMirror 복제가 구성된 경우 SSE-C 객체를 수집할 수 없습니다. 수집 작업이 실패합니다.

관련 정보

"Amazon S3 사용자 가이드: 고객 제공 키를 사용한 서버 측 암호화(SSE-C) 사용"

S3 CopyObject 요청을 사용하여 **StorageGRID**에 객체의 복사본을 생성합니다

S3 CopyObject 요청을 사용하면 S3에 이미 저장된 객체의 복사본을 만들 수 있습니다. CopyObject 작업은 GetObject 다음에 PutObject를 수행하는 것과 같습니다.

충돌 해결

두 클라이언트가 동일한 키에 쓰기를 시도하는 등 클라이언트 요청이 충돌하는 경우 "최신 요청 우선" 원칙에 따라 처리됩니다. "최신 요청 우선" 평가 시점은 S3 클라이언트가 작업을 시작한 시점이 아니라 StorageGRID 시스템이 해당 요청을 완료한 시점을 기준으로 합니다.

객체 크기

단일 PutObject 작업에 권장되는 최대 크기는 5GiB(5,368,709,120바이트)입니다. 객체 크기가 5GiB보다 큰 경우, "멀티파트 업로드"을 사용하십시오.

단일 PutObject 작업에 대해 지원되는 최대 크기는 5TiB(5,497,558,138,880바이트)입니다.

> StorageGRID 11.6 이하 버전에서 업그레이드한 경우, 5GiB를 초과하는 객체를 업로드하려고 하면 S3 PUT Object size too large 경고가 트리거됩니다. StorageGRID 11.7 또는 11.8을 새로 설치한 경우에는 이 경우 경고가 트리거되지 않습니다. 하지만 AWS S3 표준을 준수하기 위해 향후 StorageGRID 릴리스에서는 5GiB보다 큰 객체 업로드를 지원하지 않을 예정입니다.

사용자 메타데이터의 **UTF-8** 문자

요청에 사용자 정의 메타데이터의 키 이름 또는 값에 (이스케이프되지 않은) UTF-8 값이 포함된 경우 StorageGRID 동작은 정의되지 않습니다.

StorageGRID는 사용자 정의 메타데이터의 키 이름 또는 값에 포함된 이스케이프 처리된 UTF-8 문자를 구문 분석하거나 해석하지 않습니다. 이스케이프 처리된 UTF-8 문자는 ASCII 문자로 처리됩니다.

- 사용자 정의 메타데이터에 이스케이프 처리된 UTF-8 문자가 포함되어 있으면 요청이 성공합니다.
- StorageGRID는 키 이름이나 값의 해석된 값에 인쇄할 수 없는 문자가 포함된 경우 `x-amz-missing-meta` 헤더를 반환하지 않습니다.

지원되는 요청 헤더

다음과 같은 요청 헤더가 지원됩니다.

- `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-`, 그 뒤에 사용자 정의 메타데이터를 포함하는 이름-값 쌍이 옵니다
- `x-amz-metadata-directive`: 기본값은 `COPY`이며, 이를 사용하면 객체와 관련 메타데이터를 복사할 수 있습니다.

  객체를 복사할 때 기존 메타데이터를 덮어쓰거나 객체 메타데이터를 업데이트하도록 `REPLACE`을 지정할 수 있습니다.

- `x-amz-storage-class`
- `x-amz-tagging-directive`: 기본값은 `COPY`이며, 이 값을 사용하면 객체와 모든 태그를 복사할 수 있습니다.

  객체를 복사할 때 기존 태그를 덮어쓸지, 아니면 태그를 업데이트할지 지정할 수 있습니다 `REPLACE`.

- S3 객체 잠금 요청 헤더:
  - `x-amz-object-lock-mode`
  - `x-amz-object-lock-retain-until-date`
  - `x-amz-object-lock-legal-hold`

    이러한 헤더 없이 요청이 이루어지면 버킷의 기본 보존 설정이 객체 버전 모드와 보존 만료일을 계산하는 데 사용됩니다. "S3 REST API를 사용하여 S3 Object Lock 구성"을 참조하십시오.

- SSE 요청 헤더:
  - `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`

    서버 측 암호화에 대한 요청 헤더을 참조하십시오

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않습니다.

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

  `If-Match header`은(는) 승인되었지만 작동하지 않습니다.

- If-None-Match

  `If-None-Match header`은(는) 승인되었지만 작동하지 않습니다.

- x-amz-checksum-algorithm

  객체를 복사할 때 원본 객체에 체크섬이 있는 경우 StorageGRID는 해당 체크섬 값을 새 객체에 복사하지 않습니다. 이 동작은 객체 요청에서 `x-amz-checksum-algorithm`을(를) 사용하려고 시도하는지 여부와 관계없이 적용됩니다.

- x-amz-website-redirect-location

스토리지 클래스 옵션

`x-amz-storage-class` 요청 헤더는 지원되며, 일치하는 ILM 규칙에서 이중 커밋 또는 균형 xref:{relative_path}../ilm/data-protection-options-for-ingest.html["수집 옵션"]을 사용하는 경우 StorageGRID가 생성하는 객체 복사본 수에 영향을 미칩니다.

- STANDARD

  (기본값) ILM 규칙에서 이중 커밋 옵션을 사용하거나 균형 옵션이 임시 복사본 생성으로 대체될 때 이중 커밋 수집 작업을 지정합니다.

- REDUCED_REDUNDANCY

  ILM 규칙에서 이중 커밋 옵션을 사용하거나 균형 옵션이 임시 복사본 생성으로 대체될 때 단일 커밋 수집 작업을 지정합니다.

  > ⓘ S3 객체 잠금이 활성화된 버킷에 객체를 수집하는 경우 REDUCED_REDUNDANCY 옵션은 무시됩니다. 레거시 규정 준수 버킷에 객체를 수집하는 경우 REDUCED_REDUNDANCY 옵션은 오류를 반환합니다. StorageGRID는 규정 준수 요구 사항을 충족하기 위해 항상 이중 커밋 수집을 수행합니다.

CopyObject에서 x-amz-copy-source 사용

`x-amz-copy-source` 헤더에 지정된 소스 버킷 및 키가 대상 버킷 및 키와 다른 경우, 소스 객체 데이터의 복사본이 대상에 기록됩니다.

소스와 대상이 일치하고 `x-amz-metadata-directive`헤더가 `REPLACE`로 지정된 경우, 객체의 메타데이터는 요청에 제공된 메타데이터 값으로 업데이트됩니다. 이 경우 StorageGRID는 객체를 다시 수집하지 않습니다. 이는 두 가지 중요한 결과를 가져옵니다.

- CopyObject를 사용하여 기존 객체를 제자리에서 암호화하거나 기존 객체의 암호화를 제자리에서 변경할 수 없습니다. `x-amz-server-side-encryption` 헤더 또는 `x-amz-server-side-encryption-customer-algorithm` 헤더를 제공하면 StorageGRID가 요청을 거부하고 `XNotImplemented`을 반환합니다.
- 일치하는 ILM 규칙에 지정된 수집 동작 옵션은 사용되지 않습니다. 업데이트로 인해 발생하는 객체 배치 변경 사항은 일반적인 백그라운드 ILM 프로세스에 의해 ILM이 재평가될 때 적용됩니다.

  즉, ILM 규칙에서 수집 동작에 대해 엄격(Strict) 옵션을 사용하는 경우, 필요한 객체 배치가 불가능한 경우(예: 새로 필요한 위치를 사용할 수 없는 경우) 아무런 조치도 취해지지 않습니다. 업데이트된 객체는 필요한 배치가 가능해질 때까지 현재 위치를 유지합니다.

서버 측 암호화에 대한 요청 헤더

"서버 측 암호화 사용"를 수행하는 경우, 제공하는 요청 헤더는 소스 객체가 암호화되었는지 여부와 대상 객체를 암호화할 계획인지 여부에 따라 달라집니다.

- 원본 객체가 고객 제공 키(SSE-C)를 사용하여 암호화된 경우, 객체를 복호화한 후 복사할 수 있도록 CopyObject 요청에 다음 세 가지 헤더를 포함해야 합니다.
  - `x-amz-copy-source-server-side-encryption-customer-algorithm`: `AES256`을(를) 지정합니다.
  - `x-amz-copy-source-server-side-encryption-customer-key`: 소스 객체를 생성할 때 제공한 암호화 키를 지정하십시오.
  - `x-amz-copy-source-server-side-encryption-customer-key-MD5`: 소스 객체를 생성할 때 제공한 MD5 다이제스트를 지정하십시오.
- 대상 객체(복사본)를 사용자가 제공하고 관리하는 고유 키로 암호화하려면 다음 세 가지 헤더를 포함하십시오.
  - `x-amz-server-side-encryption-customer-algorithm`: `AES256`을(를) 지정합니다.
  - `x-amz-server-side-encryption-customer-key`: 대상 객체에 사용할 새 암호화 키를 지정합니다.
  - `x-amz-server-side-encryption-customer-key-MD5`: 새 암호화 키의 MD5 다이제스트를 지정합니다.

  > 고객이 제공하는 암호화 키는 절대 저장되지 않습니다. 암호화 키를 분실하면 해당 객체도 손실됩니다. 고객이 제공한 키를 사용하여 객체 데이터를 보호하기 전에 "서버 측 암호화 사용"에 대한 고려 사항을 검토하십시오.
- StorageGRID(SSE)에서 관리하는 고유 키로 대상 객체(복사본)를 암호화하려면 CopyObject 요청에 다음 헤더를 포함하십시오.
  - `x-amz-server-side-encryption`

The `server-side-encryption` 객체의 값은 업데이트할 수 없습니다. 대신, 새 `server-side-encryption` 값으로 복사본을 만들려면 `x-amz-metadata-directive`: `REPLACE`을 사용하세요.

버전 관리

원본 버킷에 버전 관리 기능이 있는 경우 `x-amz-copy-source`헤더를 사용하여 객체의 최신 버전을 복사할 수 있습니다. 특정 버전의 객체를 복사하려면 `versionId`하위 리소스를 사용하여 복사할 버전을 명시적으로 지정해야 합니다. 대상 버킷에 버전 관리 기능이 있는 경우 생성된 버전이 `x-amz-version-id`응답 헤더에 반환됩니다. 대상 버킷의 버전 관리가 일시 중단된 경우 `x-amz-version-id`"null" 값이 반환됩니다.

GetObject 요청을 사용하여 **StorageGRID**의 버킷에서 객체 검색

S3 GetObject 요청을 사용하여 S3 버킷에서 객체를 검색할 수 있습니다.

GetObject 및 멀티파트 객체

You can use the `partNumber` 요청 매개변수를 사용하여 여러 부분으로 구성된 객체 또는 분할된 객체의 특정 부분을 검색할 수 있습니다. `x-amz-mp-parts-count`응답 요소는 객체가 몇 부분으로 구성되어 있는지를 나타냅니다.

`partNumber`을 분할/멀티파트 객체와 비분할/비멀티파트 객체 모두에 대해 `1`로 설정할 수 있지만, `x-amz-mp-parts-count` 응답 요소는 분할 또는 멀티파트 객체에 대해서만 반환됩니다.

사용자 메타데이터의 **UTF-8** 문자

StorageGRID는 사용자 정의 메타데이터에서 이스케이프된 UTF-8 문자를 구문 분석하거나 해석하지 않습니다. 사용자 정의 메타데이터에 이스케이프된 UTF-8 문자가 있는 객체에 대한 GET 요청은 키 이름이나 값에 인쇄할 수 없는 문자가 포함된 경우 `x-amz-missing-meta` 헤더를 반환하지 않습니다.

지원되는 요청 헤더

다음 요청 헤더가 지원됩니다.

- `x-amz-checksum-mode`: 지정 `ENABLED`

  `Range` 헤더는 `GetObject`에 대해 `x-amz-checksum-mode`와(과) 함께 지원되지 않습니다. 요청에 `Range`을(를) 포함하고 `x-amz-checksum-mode`이(가) 활성화된 경우, `StorageGRID`는 응답에서 체크섬 값을 반환하지 않습니다.

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않으며 `XNotImplemented`을(를) 반환합니다.

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

버전 관리

If a `versionId` 하위 리소스가 지정되지 않은 경우, 해당 작업은 버전이 지정된 버킷에서 객체의 최신 버전을 가져옵니다. 현재 객체 버전이 삭제 마커인 경우, "찾을 수 없음" 상태가 반환되며 `x-amz-delete-marker` 응답 헤더가 `true`으로 설정됩니다.

고객이 제공한 암호화 키를 사용한 서버 측 암호화**(SSE-C)** 요청 헤더

제공된 고유 키로 객체가 암호화된 경우 세 가지 헤더를 모두 사용하십시오.

- `x-amz-server-side-encryption-customer-algorithm`: `AES256`을(를) 지정합니다.
- `x-amz-server-side-encryption-customer-key`: 객체에 사용할 암호화 키를 지정하십시오.
- `x-amz-server-side-encryption-customer-key-MD5`: 객체의 암호화 키의 MD5 다이제스트를 지정합니다.

> 고객이 제공하는 암호화 키는 절대 저장되지 않습니다. 암호화 키를 분실하면 해당 객체도 손실됩니다. 고객이 제공한 키를 사용하여 객체 데이터를 보호하기 전에 "서버 측 암호화 사용"의 고려 사항을 검토하십시오.

클라우드 스토리지 풀 객체에 대한 **GetObject**의 동작

객체가 "클라우드 스토리지 풀"에 저장된 경우, GetObject 요청의 동작은 객체의 상태에 따라 달라집니다. 자세한 내용은 "HeadObject"을 참조하십시오.

> 객체가 클라우드 스토리지 풀에 저장되어 있고 해당 객체의 복사본이 그리드에도 하나 이상 존재하는 경우, GetObject 요청은 클라우드 스토리지 풀에서 데이터를 가져오기 전에 그리드에서 데이터를 가져오려고 시도합니다.

객체의 상태	**GetObject**의 동작
StorageGRID에 수집되었지만 ILM에서 아직 평가되지 않은 객체이거나, 기존 스토리지 풀에 저장되었거나 이레이저 코딩을 사용하여 저장된 객체입니다.	`200 OK` 객체의 복사본이 검색됩니다.
클라우드 스토리지 풀에 있는 객체이지만 아직 검색 불가능한 상태로 전환되지 않았습니다.	`200 OK` 객체의 복사본이 검색됩니다.
객체가 복구 불가능한 상태로 전환되었습니다.	`403 Forbidden, InvalidObjectState` "RestoreObject" 요청을 사용하여 객체를 검색 가능한 상태로 복원하십시오.
복구 불가능한 상태에서 복원 중인 객체	`403 Forbidden, InvalidObjectState` RestoreObject 요청이 완료될 때까지 기다립니다.

객체의 상태	**GetObject**의 동작
객체가 클라우드 스토리지 풀에 완전히 복원되었습니다.	`200 OK` 객체의 복사본이 검색됩니다.

클라우드 스토리지 풀의 멀티파트 또는 분할된 객체

여러 부분으로 나뉜 객체를 업로드했거나 StorageGRID가 큰 객체를 여러 세그먼트로 분할한 경우, StorageGRID는 객체의 일부 또는 세그먼트를 샘플링하여 클라우드 스토리지 풀에서 해당 객체를 사용할 수 있는지 여부를 확인합니다. 경우에 따라 GetObject 요청이 객체의 일부가 이미 복구 불가능한 상태로 전환되었거나 아직 복원되지 않은 경우 잘못된 결과를 반환할 수 있습니다. `200 OK`

이러한 경우:

- GetObject 요청은 일부 데이터를 반환할 수 있지만 전송 도중에 중단될 수 있습니다.
- 이후의 GetObject 요청은 `403 Forbidden`을 반환할 수 있습니다.

GetObject 및 그리드 간 복제

"그리드 페더레이션" 및 "교차 그리드 복제"가 버킷에 대해 활성화된 경우, S3 클라이언트는 GetObject 요청을 통해 객체의 복제 상태를 확인할 수 있습니다. 응답에는 StorageGRID 관련 `x-ntap-sg-cgr-replication-status` 응답 헤더가 포함되며, 이 헤더에는 다음 값 중 하나가 있습니다.

그리드	복제 상태
소스	• 완료됨: 복제가 성공적으로 완료되었습니다. • **PENDING**: 해당 개체가 아직 복제되지 않았습니다. • **FAILURE**: 복제가 영구적으로 실패했습니다. 사용자가 오류를 해결해야 합니다.
목적지	**REPLICA**: 해당 객체는 원본 그리드에서 복제되었습니다.

StorageGRID는 `x-amz-replication-status` 헤더를 지원하지 않습니다.

S3 HeadObject 요청을 사용하여 **StorageGRID**의 객체에서 메타데이터를 검색합니다

S3 HeadObject 요청을 사용하면 객체 자체를 반환하지 않고도 객체의 메타데이터를 검색할 수 있습니다. 객체가 클라우드 스토리지 풀에 저장된 경우 HeadObject를 사용하여 객체의 전환 상태를 확인할 수 있습니다.

HeadObject 및 멀티파트 객체

멀티파트 또는 분할된 객체의 특정 부분에 대한 `partNumber` 요청 매개변수를 사용하여 메타데이터를 검색할 수 있습니다. `x-amz-mp-parts-count` 응답 요소는 객체가 몇 부분으로 구성되어 있는지를 나타냅니다.

`partNumber`을 분할/멀티파트 객체와 비분할/비멀티파트 객체 모두에 대해 1로 설정할 수 있지만, `x-amz-mp-parts-count` 응답 요소는 분할 또는 멀티파트 객체에 대해서만 반환됩니다.

사용자 메타데이터의 **UTF-8** 문자

StorageGRID는 사용자 정의 메타데이터에서 이스케이프된 UTF-8 문자를 구문 분석하거나 해석하지 않습니다. 사용자 정의 메타데이터에 이스케이프된 UTF-8 문자가 있는 객체에 대한 HEAD 요청은 키 이름이나 값에 인쇄할 수 없는 문자가 포함된 경우 `x-amz-missing-meta` 헤더를 반환하지 않습니다.

지원되는 요청 헤더

다음 요청 헤더가 지원됩니다.

- `x-amz-checksum-mode`

  `partNumber` 매개변수와 `Range` 헤더는 `HeadObject`에 대한 `x-amz-checksum-mode`에서 지원되지 않습니다. 해당 매개변수와 헤더를 `x-amz-checksum-mode`가 활성화된 요청에 포함하면 `StorageGRID`는 응답에 체크섬 값을 반환하지 않습니다.

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않으며 `XNotImplemented`을(를) 반환합니다.

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

버전 관리

If a `versionId` 하위 리소스가 지정되지 않은 경우, 해당 작업은 버전이 지정된 버킷에서 객체의 최신 버전을 가져옵니다. 현재 객체 버전이 삭제 마커인 경우, "찾을 수 없음" 상태가 반환되며 `x-amz-delete-marker` 응답 헤더가 `true`으로 설정됩니다.

고객이 제공한 암호화 키를 사용한 서버 측 암호화**(SSE-C)** 요청 헤더

제공된 고유 키로 객체가 암호화된 경우 이 세 가지 헤더를 모두 사용하십시오.

- `x-amz-server-side-encryption-customer-algorithm`: `AES256`을(를) 지정합니다.
- `x-amz-server-side-encryption-customer-key`: 객체에 사용할 암호화 키를 지정하십시오.
- `x-amz-server-side-encryption-customer-key-MD5`: 객체의 암호화 키의 MD5 다이제스트를 지정합니다.

> ⚠ 고객이 제공하는 암호화 키는 절대 저장되지 않습니다. 암호화 키를 분실하면 해당 객체도 손실됩니다. 고객이 제공한 키를 사용하여 객체 데이터를 보호하기 전에 "서버 측 암호화 사용"의 고려 사항을 검토하십시오.

HeadObject 클라우드 스토리지 풀 객체에 대한 응답

객체가 "클라우드 스토리지 풀"에 저장된 경우 다음과 같은 응답 헤더가 반환됩니다.

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

응답 헤더는 객체가 클라우드 스토리지 풀로 이동될 때, 선택적으로 복구 불가능한 상태로 전환될 때, 그리고 복원될 때 객체의 상태에 대한 정보를 제공합니다.

객체의 상태	**HeadObject**에 대한 응답
StorageGRID에 수집되었지만 ILM에서 아직 평가되지 않은 객체이거나, 기존 스토리지 풀에 저장되었거나 이레이저 코딩을 사용하여 저장된 객체입니다.	`200 OK` (특별한 응답 헤더는 반환되지 않습니다.)

클라우드 스토리지 풀에 있는 객체이지만 아직 검색 불가능한 상태로 전환되지 않았습니다.	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="false", expiry-date="Sat, 23 July 20 2030 00:00:00 GMT"`
	
	객체가 검색 불가능한 상태로 전환될 때까지 `expiry-date`의 값은 먼 미래의 시간으로 설정됩니다. 전환 시점은 StorageGRID 시스템에서 제어하지 않습니다.

객체가 검색할 수 없는 상태로 전환되었지만, 그리드에 적어도 하나의 복사본이 존재합니다.	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="false", expiry-date="Sat, 23 July 20 2030 00:00:00 GMT"`
	
	해당 값은 `expiry-date` 먼 미래의 어느 시점으로 설정되어 있습니다.
	
	참고: 그리드의 복사본을 사용할 수 없는 경우(예: 스토리지 노드가 다운된 경우) 객체를 성공적으로 검색하려면 클라우드 스토리지 풀에서 복사본을 복원하기 위한 "RestoreObject" 요청을 실행해야 합니다.

객체가 검색할 수 없는 상태로 전환되었으며, 그리드에 복사본이 존재하지 않습니다.	`200 OK`
	
	`x-amz-storage-class: GLACIER`

객체의 상태	**HeadObject**에 대한 응답

복구 불가능한 상태에서 복원 중인 객체	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="true"`

객체가 클라우드 스토리지 풀에 완전히 복원되었습니다.	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	`x-amz-restore: ongoing-request="false", expiry-date="Sat, 23 July 20 2018 00:00:00 GMT"`
	
	`expiry-date`는 Cloud Storage Pool의 객체가 검색 불가능한 상태로 반환되는 시점을 나타냅니다.

클라우드 스토리지 풀의 멀티파트 또는 분할된 객체

여러 부분으로 나뉜 객체를 업로드했거나 StorageGRID가 큰 객체를 여러 세그먼트로 분할한 경우, StorageGRID는 객체의 일부 또는 세그먼트를 샘플링하여 클라우드 스토리지 풀에서 해당 객체를 사용할 수 있는지 여부를 확인합니다. 경우에 따라 HeadObject 요청이 객체의 일부가 이미 복구 불가능한 상태로 전환되었거나 아직 복원되지 않은 경우에도 잘못된 결과를 `x-amz-restore: ongoing-request="false"` 반환할 수 있습니다.

HeadObject 및 그리드 간 복제

"그리드 페더레이션"를 사용하고 버킷에 대해 "교차 그리드 복제"가 활성화된 경우, S3 클라이언트는 HeadObject 요청을 통해 객체의 복제 상태를 확인할 수 있습니다. 응답에는 StorageGRID 관련 `x-ntap-sg-cgr-replication-status` 응답 헤더가 포함되며, 이 헤더에는 다음 값 중 하나가 있습니다.

그리드	복제 상태
소스	• 완료됨: 복제가 성공적으로 완료되었습니다. • **PENDING**: 해당 개체가 아직 복제되지 않았습니다. • **FAILURE**: 복제가 영구적으로 실패했습니다. 사용자가 오류를 해결해야 합니다.
목적지	**REPLICA**: 해당 객체는 원본 그리드에서 복제되었습니다.

StorageGRID는 `x-amz-replication-status` 헤더를 지원하지 않습니다.

S3 PutObject 요청을 사용하여 버킷에 객체를 추가할 수 있습니다.

충돌 해결

두 클라이언트가 동일한 키에 쓰기를 시도하는 등 클라이언트 요청이 충돌하는 경우 "최신 요청 우선" 원칙에 따라 처리됩니다. "최신 요청 우선" 평가 시점은 S3 클라이언트가 작업을 시작한 시점이 아니라 StorageGRID 시스템이 해당 요청을 완료한 시점을 기준으로 합니다.

객체 크기

단일 PutObject 작업에 권장되는 최대 크기는 5GiB(5,368,709,120바이트)입니다. 객체 크기가 5GiB보다 큰 경우, "멀티파트 업로드"을 사용하십시오.

단일 PutObject 작업에 대해 지원되는 최대 크기는 5TiB(5,497,558,138,880바이트)입니다.

> StorageGRID 11.6 이하 버전에서 업그레이드한 경우, 5GiB를 초과하는 객체를 업로드하려고 하면 S3 PUT Object size too large 경고가 트리거됩니다. StorageGRID 11.7 또는 11.8을 새로 설치한 경우에는 이 경우 경고가 트리거되지 않습니다. 하지만 AWS S3 표준을 준수하기 위해 향후 StorageGRID 릴리스에서는 5GiB보다 큰 객체 업로드를 지원하지 않을 예정입니다.

사용자 메타데이터 크기

Amazon S3는 각 PUT 요청 헤더 내 사용자 정의 메타데이터의 크기를 2KB로 제한합니다. StorageGRID는 사용자 메타데이터를 24KiB로 제한합니다. 사용자 정의 메타데이터의 크기는 각 키와 값의 UTF-8 인코딩 바이트 수를 합산하여 측정합니다.

사용자 메타데이터의 **UTF-8** 문자

요청에 사용자 정의 메타데이터의 키 이름 또는 값에 (이스케이프되지 않은) UTF-8 값이 포함된 경우 StorageGRID 동작은 정의되지 않습니다.

StorageGRID는 사용자 정의 메타데이터의 키 이름 또는 값에 포함된 이스케이프 처리된 UTF-8 문자를 구문 분석하거나 해석하지 않습니다. 이스케이프 처리된 UTF-8 문자는 ASCII 문자로 처리됩니다.

- PutObject, CopyObject, GetObject, and HeadObject 요청은 사용자 정의 메타데이터에 이스케이프된 UTF-8 문자가 포함된 경우 성공합니다.
- StorageGRID는 키 이름이나 값의 해석된 값에 인쇄할 수 없는 문자가 포함된 경우 `x-amz-missing-meta` 헤더를 반환하지 않습니다.

객체 태그 제한

새 객체를 업로드할 때 태그를 추가하거나 기존 객체에 태그를 추가할 수 있습니다. StorageGRID와 Amazon S3 모두 객체당 최대 10개의 태그를 지원합니다. 객체와 연결된 태그는 고유한 태그 키를 가져야 합니다. 태그 키는 최대 128자의 유니코드 문자, 태그 값은 최대 256자의 유니코드 문자까지 사용할 수 있습니다. 키와 값은 대소문자를 구분합니다.

객체 소유권

StorageGRID에서 모든 객체는 버킷 소유자 계정이 소유합니다. 여기에는 소유자가 아닌 계정이나 익명 사용자가 생성한 객체도 포함됩니다.

다음과 같은 요청 헤더가 지원됩니다.

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

  `aws-chunked`을(를) ``Content-Encoding``에 대해 지정하면 StorageGRID 는 다음 항목을 확인하지 않습니다.

  - StorageGRID는 청크 데이터에 대해 `chunk-signature`의 유효성을 검사하지 않습니다.
  - StorageGRID는 사용자가 제공하는 값을 `x-amz-decoded-content-length` 객체와 비교하여 검증하지 않습니다.
- `Content-Language`
- `Content-Length`
- `Content-MD5`
- `Content-Type`
- `Expires`
- `Transfer-Encoding`

  청크 전송 인코딩은 `aws-chunked` 페이로드 서명도 함께 사용되는 경우에 지원됩니다.
- `x-amz-checksum-sha256`
- `x-amz-meta-`, 그 뒤에 사용자 정의 메타데이터를 포함하는 이름-값 쌍이 옵니다.

  사용자 정의 메타데이터의 이름-값 쌍을 지정할 때 다음 일반 형식을 사용하십시오.

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

  ILM 규칙의 참조 시간으로 사용자 정의 생성 시간 옵션을 사용하려면 `creation-time`을(를) 객체가 생성된 시점을 기록하는 메타데이터의 이름으로 사용해야 합니다. 예를 들면 다음과 같습니다.

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

  `creation-time`의 값은 1970년 1월 1일 이후 경과된 초 단위로 평가됩니다.

  ILM 규칙은 참조 시간에 대해 *사용자 정의 생성 시간*을 사용하면서 동시에 균형 또는 엄격 수집 옵션을 사용할 수 없습니다. ILM 규칙을 생성할 때 오류가 반환됩니다.

- `x-amz-tagging`
- S3 객체 잠금 요청 헤더
  - `x-amz-object-lock-mode`
  - `x-amz-object-lock-retain-until-date`
  - `x-amz-object-lock-legal-hold`

    이러한 헤더 없이 요청이 이루어지면 버킷의 기본 보존 설정이 객체 버전 모드와 보존 만료일을 계산하는 데 사용됩니다. "S3 REST API를 사용하여 S3 Object Lock 구성"을 참조하십시오.
- SSE 요청 헤더:
  - `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`

    서버 측 암호화에 대한 요청 헤더을 참조하십시오

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않습니다.

- `If-Match`

  `If-Match header`은(는) 승인되었지만 작동하지 않습니다.
- `If-None-Match`

  `If-None-Match header`은(는) 승인되었지만 작동하지 않습니다.
- `x-amz-acl`
- `x-amz-sdk-checksum-algorithm`
- `x-amz-trailer`
- `x-amz-website-redirect-location`

  The `x-amz-website-redirect-location` 헤더는 `XNotImplemented`을 반환합니다.

스토리지 클래스 옵션

`x-amz-storage-class` 요청 헤더가 지원됩니다. `x-amz-storage-class`에 제출된 값은 StorageGRID 수집 중에 객체 데이터를 보호하는 방식에 영향을 미치며, StorageGRID 시스템에 객체의 영구 복사본이 몇 개 저장되는지(ILM에 의해 결정됨)에는 영향을 미치지 않습니다.

수집된 객체와 일치하는 ILM 규칙이 엄격한 수집 옵션을 사용하는 경우 x-amz-storage-class 헤더는 아무런 효과가 없습니다.

다음 값을 사용할 수 있습니다 x-amz-storage-class:

- STANDARD (기본값)
  - 듀얼 커밋: ILM 규칙에서 수집 동작에 대해 듀얼 커밋 옵션을 지정한 경우, 객체가 수집되는 즉시 해당 객체의 두 번째 복사본이 생성되어 다른 스토리지 노드에 배포됩니다(듀얼 커밋). ILM을 평가할 때 StorageGRID는 이러한 초기 임시 복사본이 규칙의 배치 지침을 충족하는지 확인합니다. 충족하지 않는 경우, 다른 위치에 새 객체 복사본을 생성하고 초기 임시 복사본을 삭제해야 할 수 있습니다.
  - 균형: ILM 규칙에 균형 옵션이 지정되어 있고 StorageGRID가 규칙에 지정된 모든 복사본을 즉시 생성할 수 없는 경우 StorageGRID는 서로 다른 스토리지 노드에 두 개의 임시 복사본을 생성합니다.

    StorageGRID가 ILM 규칙에 지정된 모든 객체 복사본을 즉시 생성할 수 있는 경우(동기 배치), x-amz-storage-class 헤더는 아무런 효과가 없습니다.

- REDUCED_REDUNDANCY
  - 듀얼 커밋: ILM 규칙에서 수집 동작에 대해 듀얼 커밋 옵션을 지정한 경우, StorageGRID는 객체가 수집될 때 단일 임시 복사본을 생성합니다(싱글 커밋).
  - 균형: ILM 규칙에 균형 옵션이 지정된 경우, StorageGRID 시스템이 규칙에 지정된 모든 복사본을 즉시 생성할 수 없는 경우에만 임시 복사본을 하나만 생성합니다. StorageGRID 동기 배치를 수행할 수 있는 경우에는 이 헤더가 아무런 효과가 없습니다. 이 REDUCED_REDUNDANCY 옵션은 객체와 일치하는 ILM 규칙이 단일 복제본을 생성하는 경우에 가장 적합합니다. 이 경우 `REDUCED_REDUNDANCY`를 사용하면 모든 수집 작업마다 불필요한 객체 복사본 생성 및 삭제를 방지할 수 있습니다.

  다른 상황에서는 REDUCED_REDUNDANCY 옵션을 사용하는 것을 권장하지 않습니다. `REDUCED_REDUNDANCY`은(는) 수집 중 오브젝트 데이터 손실 위험을 증가시킵니다. 예를 들어, ILM 평가가 수행되기 전에 장애가 발생한 Storage Node에 단일 복사본이 처음에 저장되는 경우 데이터를 손실할 수 있습니다.

> 특정 기간 동안 복제본이 하나만 존재하는 경우 데이터가 영구적으로 손실될 위험이 있습니다. 객체의 복제본이 하나만 있는 경우 스토리지 노드에 장애가 발생하거나 중대한 오류가 발생하면 해당 객체가 손실됩니다. 또한 업그레이드와 같은 유지 관리 절차 중에도 해당 객체에 대한 액세스가 일시적으로 제한됩니다.

이 설정을 지정하면 REDUCED_REDUNDANCY 객체가 처음 수집될 때 생성되는 복사본 수에만 영향을 미칩니다. 활성 ILM 정책에 따라 객체를 평가할 때 생성되는 객체 복사본 수에는 영향을 미치지 않으며, StorageGRID 시스템에서 데이터가 더 낮은 수준의 중복성으로 저장되는 결과를 초래하지도 않습니다.

> S3 객체 잠금이 활성화된 버킷에 객체를 수집하는 경우 REDUCED_REDUNDANCY 옵션은 무시됩니다. 레거시 규정 준수 버킷에 객체를 수집하는 경우 REDUCED_REDUNDANCY 옵션은 오류를 반환합니다. StorageGRID는 규정 준수 요구 사항을 충족하기 위해 항상 이중 커밋 수집을 수행합니다.

서버 측 암호화에 대한 요청 헤더

다음 요청 헤더를 사용하여 서버 측 암호화로 객체를 암호화할 수 있습니다. SSE와 SSE-C 옵션은 상호 배타적입니다.

- **SSE**: StorageGRID에서 관리하는 고유 키로 객체를 암호화하려면 다음 헤더를 사용하십시오.
  - `x-amz-server-side-encryption`

    `x-amz-server-side-encryption` 헤더가 `PutObject` 요청에 포함되지 않으면 grid-wide xref:{relative_path}../admin/changing-network-options-object-encryption.html["저장된 객체 암호화 설정"]가 `PutObject` 응답에서 생략됩니다.

- **SSE-C**: 직접 제공하고 관리하는 고유 키로 객체를 암호화하려면 이 세 가지 헤더를 모두 사용하십시오.
  - `x-amz-server-side-encryption-customer-algorithm`: `AES256`을(를) 지정합니다.
  - `x-amz-server-side-encryption-customer-key`: 새 객체에 사용할 암호화 키를 지정하십시오.
  - `x-amz-server-side-encryption-customer-key-MD5`: 새 객체의 암호화 키의 MD5 다이제스트를 지정합니다.

고객이 제공하는 암호화 키는 절대 저장되지 않습니다. 암호화 키를 분실하면 해당 객체도 손실됩니다. 고객이 제공한 키를 사용하여 객체 데이터를 보호하기 전에 "서버 측 암호화 사용"에 대한 고려 사항을 검토하십시오.

객체가 SSE 또는 SSE-C로 암호화된 경우 버킷 수준 또는 그리드 수준의 암호화 설정은 무시됩니다.

버전 관리

버킷에 버전 관리가 활성화된 경우, 저장된 객체의 버전에 대한 고유 `versionId`가 자동으로 생성됩니다. 이 `versionId`는 `x-amz-version-id`응답 헤더를 사용하여 응답에도 반환됩니다.

버전 관리가 일시 중단되면 객체 버전은 null `versionId`으로 저장되며, 이미 null 버전이 존재하는 경우 덮어쓰여집니다.

Authorization 헤더에 대한 서명 계산

`Authorization` 헤더를 사용하여 요청을 인증할 때 `StorageGRID`는 다음과 같은 점에서 `AWS`와 다릅니다.

- StorageGRID는 헤더를 `host` 내에 포함할 필요가 없습니다 `CanonicalHeaders`.
- StorageGRID는 `Content-Type`이(가) `CanonicalHeaders` 내에 포함될 필요가 없습니다.
- StorageGRID는 `x-amz-*` 헤더가 `CanonicalHeaders` 안에 포함될 필요가 없습니다.

일반적으로 이러한 헤더를 항상 `CanonicalHeaders` 내에 포함하여 검증하는 것이 좋습니다. 하지만 이러한 헤더를 제외하더라도 StorageGRID는 오류를 반환하지 않습니다.

자세한 내용은 "권한 부여 헤더에 대한 서명 계산: 단일 청크로 페이로드 전송(AWS Signature Version 4)"를

참조하십시오.

관련 정보

- "ILM을 사용하여 객체 관리"
- "Amazon Simple Storage Service API 참조: PutObject"

S3 RestoreObject 요청을 사용하여 **StorageGRID**에서 객체 복원

S3 RestoreObject 요청을 사용하면 클라우드 스토리지 풀에 저장된 객체를 복원할 수 있습니다.

지원되는 요청 유형

StorageGRID는 객체를 복원하기 위해 RestoreObject 요청만 지원합니다. `SELECT` 복원 유형은 지원하지 않습니다. Select 요청은 `XNotImplemented`을 반환합니다.

버전 관리

선택적으로, `versionId`버전이 지정된 버킷에서 객체의 특정 버전을 복원하도록 지정할 수 있습니다. 지정하지 않으면 `versionId`객체의 최신 버전이 복원됩니다.

클라우드 스토리지 풀 객체에 대한 **RestoreObject**의 동작

객체가 "클라우드 스토리지 풀"에 저장된 경우, RestoreObject 요청은 객체의 상태에 따라 다음과 같은 동작을 합니다. 자세한 내용은 "HeadObject"을 참조하십시오.

> 객체가 클라우드 스토리지 풀에 저장되어 있고 해당 객체의 복사본이 그리드에도 하나 이상 존재하는 경우, RestoreObject 요청을 실행하여 객체를 복원할 필요가 없습니다. 대신 GetObject 요청을 사용하여 로컬 복사본을 직접 검색할 수 있습니다.

객체의 상태	**RestoreObject**의 동작
StorageGRID에 수집되었지만 ILM에서 아직 평가되지 않은 객체이거나, 클라우드 스토리지 풀에 없는 객체입니다.	`403 Forbidden, InvalidObjectState`
클라우드 스토리지 풀에 있는 객체이지만 아직 검색 불가능한 상태로 전환되지 않았습니다.	`200 OK` 변경 사항이 없습니다. 참고: 객체가 검색 불가능한 상태로 전환되기 전에는 해당 객체의 `expiry-date`을(를) 변경할 수 없습니다.

객체의 상태	**RestoreObject**의 동작
객체가 복구 불가능한 상태로 전환되었습니다.	`202 Accepted` 요청 본문에 지정된 일수 동안 객체의 검색 가능한 복사본을 클라우드 스토리지 풀에 복원합니다. 이 기간이 종료되면 객체는 검색 불가능한 상태로 돌아갑니다. 선택적으로, `Tier` 요청 요소를 사용하여 복원 작업이 완료되는 데 걸리는 시간((`Expedited`, `Standard`, 또는 `Bulk`)을 결정할 수 있습니다. `Tier`를 지정하지 않으면 `Standard` 티어가 사용됩니다. 중요: 개체가 S3 Glacier Deep Archive로 전환되었거나 클라우드 스토리지 풀이 Azure Blob 스토리지를 사용하는 경우, 해당 `Expedited` 계층을 사용하여 복원할 수 없습니다. 다음 오류가 반환됩니다 `403 Forbidden`, `InvalidTier`: `Retrieval option is not supported by this storage class`.
복구 불가능한 상태에서 복원 중인 객체	`409 Conflict`, `RestoreAlreadyInProgress`
객체가 클라우드 스토리지 풀에 완전히 복원되었습니다.	`200 OK` 참고: 객체가 검색 가능한 상태로 복원된 경우, `expiry-date`에 대한 새 값으로 `RestoreObject` 요청을 다시 실행하여 복원 날짜를 변경할 수 있습니다. `Days` 복원 날짜는 요청 시간을 기준으로 업데이트됩니다.

S3 SelectObjectContent 요청을 사용하여 **StorageGRID**에서 객체 데이터 필터링

S3 SelectObjectContent 요청을 사용하면 간단한 SQL 문을 기반으로 S3 객체의 내용을 필터링할 수 있습니다.

자세한 내용은 "Amazon Simple Storage Service API 참조: SelectObjectContent"을 참조하십시오.

시작하기 전에

- 테넌트 계정에는 S3 Select 권한이 있습니다.
- 조회하려는 개체에 대한 `s3:GetObject` 권한이 있습니다.
- 조회하려는 객체는 다음 형식 중 하나여야 합니다.
  - **CSV**. 있는 그대로 사용하거나 GZIP 또는 BZIP2 아카이브로 압축할 수 있습니다.
  - **Parquet**. Parquet 객체에 대한 추가 요구사항:
    - S3 Select는 GZIP 또는 Snappy를 사용한 컬럼형 압축만 지원합니다. S3 Select는 Parquet 객체에 대한 전체 객체 압축을 지원하지 않습니다.
    - S3 Select는 Parquet 출력을 지원하지 않습니다. 출력 형식을 CSV 또는 JSON으로 지정해야 합니다.
    - 압축되지 않은 행 그룹의 최대 크기는 512MB입니다.
    - 객체의 스키마에 명시된 데이터 형식을 사용해야 합니다.
    - INTERVAL, JSON, LIST, TIME 또는 UUID 논리 유형은 사용할 수 없습니다.

- SQL 표현식의 최대 길이는 256 KB입니다.
- 입력 또는 결과의 모든 레코드는 최대 길이가 1MiB입니다.

CSV 요청 구문 예시

```
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>
```

Parquet 요청 구문 예시

```
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 쿼리 예시

이 쿼리는 주 이름, 2010년 인구, 2015년 추정 인구 및 미국 인구 조사 데이터 대비 변화율을 가져옵니다. 파일에서 주가 아닌 레코드는 무시됩니다.

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

조회할 파일의 처음 몇 줄인 `SUB-EST2020_ALL.csv`은 다음과 같습니다.

```
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 사용 예시 **(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
```

출력 파일의 처음 몇 줄인 `changes.csv`은 다음과 같습니다.

```
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 사용 예시 (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
```

출력 파일인 changes.csv의 처음 몇 줄은 다음과 같습니다.

```
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
```

멀티파트 업로드 작업

StorageGRID에서 지원되는 멀티파트 업로드 작업

이 섹션에서는 StorageGRID가 멀티파트 업로드 작업을 지원하는 방법을 설명합니다.

모든 멀티파트 업로드 작업에는 다음과 같은 조건과 유의사항이 적용됩니다.

- 단일 버킷에 대한 동시 멀티파트 업로드는 1,000개를 초과해서는 안 됩니다. 초과할 경우 해당 버킷에 대한 ListMultipartUploads 쿼리 결과가 불완전하게 반환될 수 있습니다.
- StorageGRID는 멀티파트 파트에 대한 AWS 크기 제한을 적용합니다. S3 클라이언트는 다음 지침을 준수해야 합니다.
  - 멀티파트 업로드의 각 파트는 5MiB(5,242,880바이트)에서 5GiB(5,368,709,120바이트) 사이여야 합니다.
  - 마지막 부분의 크기는 5MiB(5,242,880바이트)보다 작을 수 있습니다.
  - 일반적으로 파트 크기는 가능한 한 크게 설정하는 것이 좋습니다. 예를 들어 100GiB 객체의 경우 5GiB의 파트 크기를 사용하십시오. 각 파트는 고유한 객체로 간주되므로 큰 파트 크기를 사용하면 StorageGRID 메타데이터 오버헤드를 줄일 수 있습니다.
  - 5GiB보다 작은 객체의 경우 멀티파트가 아닌 업로드 방식을 사용하는 것을 고려하십시오.
- ILM 규칙이 균형(Balanced) 또는 엄격(Strict) "수집 옵션"을 사용하는 경우, 멀티파트 객체가 수집될 때 각 파트에 대해 ILM이 평가되고, 멀티파트 업로드가 완료될 때 객체 전체에 대해 평가됩니다. 이러한 점이 객체 및 파트 배치에 미치는 영향을 숙지해야 합니다.
  - S3 멀티파트 업로드가 진행되는 동안 ILM이 변경되면, 멀티파트 업로드가 완료될 때 객체의 일부 파트가 현재 ILM 요구 사항을 충족하지 못할 수 있습니다. 올바르게 배치되지 않은 파트는 ILM 재평가를 위해 대기열에 추가된 후 나중에 올바른 위치로 이동됩니다.

    - StorageGRID는 파트에 대한 ILM을 평가할 때 객체 전체의 크기가 아닌 파트의 크기를 기준으로 필터링합니다. 즉, 객체의 일부는 객체 전체에 대한 ILM 요구 사항을 충족하지 않는 위치에 저장될 수 있습니다. 예를 들어, 10GB 이상의 모든 객체는 DC1에 저장하고 10GB 미만의 모든 객체는 DC2에 저장하도록 규칙이 지정된 경우, 10개 파트로 구성된 멀티파트 업로드의 각 1GB 파트는 수집 시 DC2에 저장됩니다. 그러나 객체 전체에 대한 ILM이 평가되면 객체의 모든 파트가 DC1으로 이동됩니다.

- 모든 멀티파트 업로드 작업은 StorageGRID "일관성 값"를 지원합니다.
- 멀티파트 업로드를 사용하여 객체를 수집할 경우, "객체 분할 임계값(1 GiB)"이 적용되지 않습니다.
- 필요에 따라 "서버 측 암호화"멀티파트 업로드를 사용할 수 있습니다. SSE(StorageGRID에서 관리하는 키를 사용하는 서버 측 암호화)를 사용하려면 `x-amz-server-side-encryption` 요청 헤더를 CreateMultipartUpload 요청에만 포함하면 됩니다. SSE-C(고객이 제공하는 키를 사용하는 서버 측 암호화)를 사용하려면 CreateMultipartUpload 요청과 이후의 각 UploadPart 요청에 동일한 세 가지 암호화 키 요청 헤더를 지정해야 합니다.
- 멀티파트 업로드 객체는 업로드 완료 시점과 관계없이 기본 버킷의 'Before' 타임스탬프 이전에 수집이 시작된 경우 "브랜치 버킷"에 포함됩니다.

작업	구현
AbortMultipartUpload	모든 Amazon S3 REST API 동작에 맞춰 구현되었습니다. 예고 없이 변경될 수 있습니다.
CompleteMultipartUpload	"CompleteMultipartUpload"을 참조하십시오
CreateMultipartUpload (이전 명칭: Initiate Multipart Upload)	"CreateMultipartUpload"을 참조하십시오
ListMultipartUploads	"ListMultipartUploads"을 참조하십시오
ListParts	모든 Amazon S3 REST API 동작에 맞춰 구현되었습니다. 예고 없이 변경될 수 있습니다.
UploadPart	"UploadPart"을 참조하십시오
UploadPartCopy	"UploadPartCopy"을 참조하십시오

StorageGRID S3 REST API가 멀티파트 업로드를 완료하는 방법

CompleteMultipartUpload 작업은 이전에 업로드된 부분들을 조합하여 객체의 멀티파트 업로드를 완료합니다.

StorageGRID는 CompleteMultipartUpload와 함께 사용되는 `partNumber` 요청 매개변수에 대해 오름차순의 비연속적인 값을 지원합니다. 매개변수는 어떤 값으로든 시작할 수 있습니다.

충돌 해결

두 클라이언트가 동일한 키에 쓰기를 시도하는 등 클라이언트 요청이 충돌하는 경우 "최신 요청 우선" 원칙에 따라

처리됩니다. "최신 요청 우선" 평가 시점은 S3 클라이언트가 작업을 시작한 시점이 아니라 StorageGRID 시스템이 해당 요청을 완료한 시점을 기준으로 합니다.

지원되는 요청 헤더

다음과 같은 요청 헤더가 지원됩니다.

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

  `x-amz-storage-class` 헤더는 일치하는 ILM 규칙이 xref:{relative_path}../ilm/data-protection-options-for-ingest.html["이중 커밋 또는 균형 잡힌 수집 옵션"]을(를) 지정하는 경우 StorageGRID가 생성하는 객체 복사본 수에 영향을 미칩니다.

- `STANDARD`

  (기본값) ILM 규칙에서 이중 커밋 옵션을 사용하거나 균형 옵션이 임시 복사본 생성으로 대체될 때 이중 커밋 수집 작업을 지정합니다.

- `REDUCED_REDUNDANCY`

  ILM 규칙에서 이중 커밋 옵션을 사용하거나 균형 옵션이 임시 복사본 생성으로 대체될 때 단일 커밋 수집 작업을 지정합니다.

  > S3 객체 잠금이 활성화된 버킷에 객체를 수집하는 경우 `REDUCED_REDUNDANCY` 옵션은 무시됩니다. 레거시 규정 준수 버킷에 객체를 수집하는 경우 `REDUCED_REDUNDANCY` 옵션은 오류를 반환합니다. StorageGRID는 규정 준수 요구 사항을 충족하기 위해 항상 이중 커밋 수집을 수행합니다.

> 멀티파트 업로드가 15일 이내에 완료되지 않으면 해당 작업은 비활성으로 표시되고 관련 데이터는 시스템에서 모두 삭제됩니다.

> 반환되는 `ETag` 값은 데이터의 MD5 합계가 아니라, 멀티파트 객체에 대한 `ETag` 값의 Amazon S3 API 구현 방식을 따릅니다.

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않습니다.

- `If-Match`

  `If-Match header`은(는) 승인되었지만 작동하지 않습니다.

- `If-None-Match`

`If-None-Match header`은(는) 승인되었지만 작동하지 않습니다.

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

버전 관리

이 작업은 멀티파트 업로드를 완료합니다. 버킷에 버전 관리가 활성화된 경우, 멀티파트 업로드가 완료된 후 객체 버전이 생성됩니다.

버킷에 버전 관리가 활성화된 경우, 저장된 객체의 버전에 대한 고유 `versionId`가 자동으로 생성됩니다. 이 `versionId`는 `x-amz-version-id`응답 헤더를 사용하여 응답에도 반환됩니다.

버전 관리가 일시 중단되면 객체 버전은 null `versionId`으로 저장되며, 이미 null 버전이 존재하는 경우 덮어쓰여집니다.

> 버킷에 버전 관리가 활성화된 경우, 동일한 객체 키에 대해 동시에 여러 개의 멀티파트 업로드가 완료되더라도 멀티파트 업로드가 완료되면 항상 새 버전이 생성됩니다. 버킷에 버전 관리가 활성화되지 않은 경우, 멀티파트 업로드를 시작한 후 동일한 객체 키에 대해 다른 멀티파트 업로드가 먼저 시작되어 완료될 수 있습니다. 버전 관리가 되지 않는 버킷에서는 마지막으로 완료된 멀티파트 업로드가 우선 순위를 갖습니다.

복제, 알림 또는 메타데이터 알림 실패

멀티파트 업로드가 발생하는 버킷이 플랫폼 서비스용으로 구성된 경우, 관련 복제 또는 알림 작업이 실패하더라도 멀티파트 업로드는 성공합니다.

테넌트는 객체의 메타데이터 또는 태그를 업데이트하여 복제 실패 또는 알림을 발생시킬 수 있습니다. 테넌트는 원치 않는 변경을 방지하려면 기존 값을 다시 제출할 수 있습니다.

"플랫폼 서비스 문제 해결"을 참조하십시오.

StorageGRID의 **S3 CreateMultipartUpload** 요청 헤더 및 옵션

CreateMultipartUpload(이전 명칭: Initiate Multipart Upload) 작업은 객체에 대한 멀티파트 업로드를 시작하고 업로드 ID를 반환합니다.

`x-amz-storage-class` 요청 헤더가 지원됩니다. `x-amz-storage-class`에 제출된 값은 `StorageGRID` 수집 중에 객체 데이터를 보호하는 방식에 영향을 미치며, `StorageGRID` 시스템에 객체의 영구 복사본이 몇 개 저장되는지(`ILM`에 의해 결정됨)에는 영향을 미치지 않습니다.

수집된 객체와 일치하는 ILM 규칙이 Strict "수집 옵션"를 사용하는 경우, `x-amz-storage-class` 헤더는 아무런 효과가 없습니다.

다음 값을 사용할 수 있습니다 `x-amz-storage-class`:

- `STANDARD` (기본값)
  - 듀얼 커밋: ILM 규칙에 듀얼 커밋 수집 옵션이 지정된 경우, 객체가 수집되는 즉시 해당 객체의 두 번째 복사본이

생성되어 다른 스토리지 노드에 배포됩니다(듀얼 커밋). ILM을 평가할 때 StorageGRID는 이러한 초기 임시 복사본이 규칙의 배치 지침을 충족하는지 확인합니다. 충족하지 않는 경우, 다른 위치에 새 객체 복사본을 생성하고 초기 임시 복사본을 삭제해야 할 수 있습니다.

  - 균형: ILM 규칙에 균형 옵션이 지정되어 있고 StorageGRID가 규칙에 지정된 모든 복사본을 즉시 생성할 수 없는 경우 StorageGRID는 서로 다른 스토리지 노드에 두 개의 임시 복사본을 생성합니다.

    StorageGRID가 ILM 규칙에 지정된 모든 객체 복사본을 즉시 생성할 수 있는 경우(동기 배치), `x-amz-storage-class` 헤더는 아무런 효과가 없습니다.

- `REDUCED_REDUNDANCY`

  - 듀얼 커밋: ILM 규칙에서 듀얼 커밋 옵션을 지정하면 StorageGRID는 객체가 수집될 때 단일 임시 복사본을 생성합니다(싱글 커밋).
  - 균형: ILM 규칙에 균형 옵션이 지정된 경우, StorageGRID 시스템이 규칙에 지정된 모든 복사본을 즉시 생성할 수 없는 경우에만 임시 복사본을 하나만 생성합니다. StorageGRID 동기 배치를 수행할 수 있는 경우에는 이 헤더가 아무런 효과가 없습니다. 이 `REDUCED_REDUNDANCY` 옵션은 객체와 일치하는 ILM 규칙이 단일 복제본을 생성하는 경우에 가장 적합합니다. 이 경우 `REDUCED_REDUNDANCY`를 사용하면 모든 수집 작업마다 불필요한 객체 복사본 생성 및 삭제를 방지할 수 있습니다.

  다른 상황에서는 `REDUCED_REDUNDANCY` 옵션을 사용하는 것을 권장하지 않습니다. `REDUCED_REDUNDANCY`은(는) 수집 중 오브젝트 데이터 손실 위험을 증가시킵니다. 예를 들어, ILM 평가가 수행되기 전에 장애가 발생한 Storage Node에 단일 복사본이 처음에 저장되는 경우 데이터를 손실할 수 있습니다.

> 특정 기간 동안 복제본이 하나만 존재하는 경우 데이터가 영구적으로 손실될 위험이 있습니다. 객체의 복제본이 하나만 있는 경우 스토리지 노드에 장애가 발생하거나 중대한 오류가 발생하면 해당 객체가 손실됩니다. 또한 업그레이드와 같은 유지 관리 절차 중에도 해당 객체에 대한 액세스가 일시적으로 제한됩니다.

이 설정을 지정하면 `REDUCED_REDUNDANCY` 객체가 처음 수집될 때 생성되는 복사본 수에만 영향을 미칩니다. 활성 ILM 정책에 따라 객체를 평가할 때 생성되는 객체 복사본 수에는 영향을 미치지 않으며, StorageGRID 시스템에서 데이터가 더 낮은 수준의 중복성으로 저장되는 결과를 초래하지도 않습니다.

> S3 객체 잠금이 활성화된 버킷에 객체를 수집하는 경우 `REDUCED_REDUNDANCY` 옵션은 무시됩니다. 레거시 규정 준수 버킷에 객체를 수집하는 경우 `REDUCED_REDUNDANCY` 옵션은 오류를 반환합니다. StorageGRID는 규정 준수 요구 사항을 충족하기 위해 항상 이중 커밋 수집을 수행합니다.

지원되는 요청 헤더

다음과 같은 요청 헤더가 지원됩니다.

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

  현재는 `x-amz-checksum-algorithm`의 SHA256 값만 지원됩니다.

- `x-amz-meta-`, 그 뒤에 사용자 정의 메타데이터를 포함하는 이름-값 쌍이 옵니다

  사용자 정의 메타데이터의 이름-값 쌍을 지정할 때 다음 일반 형식을 사용하십시오.

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

ILM 규칙의 참조 시간으로 사용자 정의 생성 시간 옵션을 사용하려면 `creation-time`을(를) 객체가 생성된 시점을 기록하는 메타데이터의 이름으로 사용해야 합니다. 예를 들면 다음과 같습니다.

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

`creation-time`의 값은 1970년 1월 1일 이후 경과된 초 단위로 평가됩니다.

`creation-time`을(를) 사용자 정의 메타데이터로 추가하는 것은 레거시 규정 준수가 활성화된 버킷에 객체를 추가하는 경우 허용되지 않습니다. 오류가 반환됩니다.

- S3 객체 잠금 요청 헤더:
  - `x-amz-object-lock-mode`
  - `x-amz-object-lock-retain-until-date`
  - `x-amz-object-lock-legal-hold`

  이러한 헤더 없이 요청이 이루어지면 버킷의 기본 보존 설정이 사용되어 객체 버전의 보존 기간이 계산됩니다.

  "S3 REST API를 사용하여 S3 Object Lock 구성"

- SSE 요청 헤더:
  - `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`

  서버 측 암호화에 대한 요청 헤더

StorageGRID가 UTF-8 문자를 처리하는 방법에 대한 자세한 내용은 "PutObject"을 참조하십시오.

서버 측 암호화에 대한 요청 헤더

다음 요청 헤더를 사용하여 서버 측 암호화로 멀티파트 객체를 암호화할 수 있습니다. SSE와 SSE-C 옵션은 상호 배타적입니다.

- **SSE**: StorageGRID에서 관리하는 고유 키로 객체를 암호화하려면 CreateMultipartUpload 요청에 다음 헤더를 사용하십시오. UploadPart 요청에는 이 헤더를 지정하지 마십시오.
  - `x-amz-server-side-encryption`
- **SSE-C**: 제공하고 관리하는 고유 키로 객체를 암호화하려면 CreateMultipartUpload 요청(및 이후의 각

UploadPart 요청)에 이 세 가지 헤더를 모두 사용하십시오.

- `x-amz-server-side-encryption-customer-algorithm`: `AES256`을(를) 지정합니다.
- `x-amz-server-side-encryption-customer-key`: 새 객체에 사용할 암호화 키를 지정하십시오.
- `x-amz-server-side-encryption-customer-key-MD5`: 새 객체의 암호화 키의 MD5 다이제스트를 지정합니다.

> 고객이 제공하는 암호화 키는 절대 저장되지 않습니다. 암호화 키를 분실하면 해당 객체도 손실됩니다. 고객이 제공한 키를 사용하여 객체 데이터를 보호하기 전에 "서버 측 암호화 사용"에 대한 고려 사항을 검토하십시오.

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않습니다.

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

  The `x-amz-website-redirect-location` 헤더는 `XNotImplemented`을 반환합니다.

버전 관리

멀티파트 업로드는 업로드 시작, 업로드 목록 보기, 파트 업로드, 업로드된 파트 조립 및 업로드 완료를 위한 개별 작업으로 구성됩니다. CompleteMultipartUpload 작업이 수행되면 객체가 생성됩니다(해당하는 경우 버전 관리도 수행됨).

S3 ListMultipartUploads 작업 및 **StorageGRID**에서 지원되는 매개 변수

ListMultipartUploads 작업은 버킷에 대해 진행 중인 멀티파트 업로드 목록을 표시합니다.

다음 요청 매개변수가 지원됩니다.

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

버전 관리

멀티파트 업로드는 업로드 시작, 업로드 목록 보기, 파트 업로드, 업로드된 파트 조립 및 업로드 완료를 위한 개별 작업으로 구성됩니다. CompleteMultipartUpload 작업이 수행되면 객체가 생성됩니다(해당하는 경우 버전 관리도 수행됨).

StorageGRID의 S3 UploadPart 작업에 대해 지원되는 요청 헤더 및 지원되지 않는 요청 헤더

UploadPart 작업은 객체의 멀티파트 업로드에서 파트를 업로드합니다.

지원되는 요청 헤더

다음과 같은 요청 헤더가 지원됩니다.

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

서버 측 암호화에 대한 요청 헤더

암호화CreateMultipartUpload 요청에 SSE-C 암호화를 지정한 경우, 각 UploadPart 요청에 다음 요청 헤더도 포함해야 합니다.

- `x-amz-server-side-encryption-customer-algorithm`: `AES256`을(를) 지정합니다.
- `x-amz-server-side-encryption-customer-key`: CreateMultipartUpload 요청 시 제공했던 것과 동일한 암호화 키를 지정하십시오.
- `x-amz-server-side-encryption-customer-key-MD5`: CreateMultipartUpload 요청에서 제공한 것과 동일한 MD5 다이제스트를 지정하십시오.

> 고객이 제공하는 암호화 키는 절대 저장되지 않습니다. 암호화 키를 분실하면 해당 객체도 손실됩니다. 고객이 제공한 키를 사용하여 객체 데이터를 보호하기 전에 "서버 측 암호화 사용"의 고려 사항을 검토하십시오.

체크섬 CreateMultipartUpload 요청 시 SHA-256 체크섬을 지정한 경우, 각 UploadPart 요청에 다음 요청 헤더를 포함해야 합니다.

- `x-amz-checksum-sha256`: 이 부분에 대한 SHA-256 체크섬을 지정하십시오.

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않습니다.

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

버전 관리

멀티파트 업로드는 업로드 시작, 업로드 목록 보기, 파트 업로드, 업로드된 파트 조립 및 업로드 완료를 위한 개별 작업으로 구성됩니다. CompleteMultipartUpload 작업이 수행되면 객체가 생성됩니다(해당하는 경우 버전 관리도 수행됨).

S3 UploadPartCopy 작업을 사용하여 StorageGRID에 객체의 일부를 업로드합니다

UploadPartCopy 작업은 기존 객체의 데이터를 데이터 소스로 복사하여 객체의 일부를 업로드합니다.

UploadPartCopy 작업은 모든 Amazon S3 REST API 동작 방식을 따릅니다. 예고 없이 변경될 수 있습니다.

이 요청은 StorageGRID 시스템 내의 `x-amz-copy-source-range`에 지정된 객체 데이터를 읽고 씁니다.

다음과 같은 요청 헤더가 지원됩니다.

- `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`

서버 측 암호화에 대한 요청 헤더

암호화CreateMultipartUpload 요청에 SSE-C 암호화를 지정한 경우, 각 UploadPartCopy 요청에도 다음 요청 헤더를 포함해야 합니다.

- `x-amz-server-side-encryption-customer-algorithm`: `AES256`을(를) 지정합니다.
- `x-amz-server-side-encryption-customer-key`: CreateMultipartUpload 요청 시 제공했던 것과 동일한 암호화 키를 지정하십시오.
- `x-amz-server-side-encryption-customer-key-MD5`: CreateMultipartUpload 요청에서 제공한 것과 동일한 MD5 다이제스트를 지정하십시오.

원본 객체가 고객 제공 키(SSE-C)를 사용하여 암호화된 경우, 객체를 복호화한 후 복사할 수 있도록 UploadPartCopy 요청에 다음 세 가지 헤더를 포함해야 합니다.

- `x-amz-copy-source-server-side-encryption-customer-algorithm`: `AES256`을(를) 지정합니다.
- `x-amz-copy-source-server-side-encryption-customer-key`: 소스 객체를 생성할 때 제공한 암호화 키를 지정하십시오.
- `x-amz-copy-source-server-side-encryption-customer-key-MD5`: 소스 객체를 생성할 때 제공한 MD5 다이제스트를 지정하십시오.

> 고객이 제공하는 암호화 키는 절대 저장되지 않습니다. 암호화 키를 분실하면 해당 객체도 손실됩니다. 고객이 제공한 키를 사용하여 객체 데이터를 보호하기 전에 "서버 측 암호화 사용"의 고려 사항을 검토하십시오.

버전 관리

멀티파트 업로드는 업로드 시작, 업로드 목록 보기, 파트 업로드, 업로드된 파트 조립 및 업로드 완료를 위한 개별 작업으로 구성됩니다. CompleteMultipartUpload 작업이 수행되면 객체가 생성됩니다(해당하는 경우 버전 관리도 수행됨).

StorageGRID S3 REST API 오류 응답

StorageGRID 시스템은 적용 가능한 모든 표준 S3 REST API 오류 응답을 지원합니다. 또한 StorageGRID 구현은 몇 가지 사용자 지정 응답을 추가로 제공합니다.

지원되는 **S3 API** 오류 코드

이름	**HTTP** 상태
AccessDenied	403 Forbidden
BadDigest	400 잘못된 요청
BucketAlreadyExists	409 충돌
BucketNotEmpty	409 충돌
IncompleteBody	400 잘못된 요청
InternalError	500 내부 서버 오류
InvalidAccessKeyId	403 Forbidden
InvalidArgument	400 잘못된 요청
InvalidBucketName	400 잘못된 요청
InvalidBucketState	409 충돌
InvalidDigest	400 잘못된 요청
InvalidEncryptionAlgorithmError	400 잘못된 요청
InvalidPart	400 잘못된 요청
InvalidPartOrder	400 잘못된 요청
InvalidRange	416 요청한 범위가 만족스럽지 않음
InvalidRequest	400 잘못된 요청
InvalidStorageClass	400 잘못된 요청
InvalidTag	400 잘못된 요청
InvalidURI	400 잘못된 요청
KeyTooLong	400 잘못된 요청

이름	**HTTP** 상태
MalformedXML	400 잘못된 요청
MetadataTooLarge	400 잘못된 요청
MethodNotAllowed	405 허용되지 않는 메서드
MissingContentLength	411 길이가 필요함
MissingRequestBodyError	400 잘못된 요청
MissingSecurityHeader	400 잘못된 요청
NoSuchBucket	404 찾을 수 없음
NoSuchKey	404 찾을 수 없음
NoSuchUpload	404 찾을 수 없음
NotImplemented	501 구현되지 않음
NoSuchBucketPolicy	404 찾을 수 없음
ObjectLockConfigurationNotFoundError	404 찾을 수 없음
PreconditionFailed	412 사전 조건 실패
RequestTimeTooSkewed	403 Forbidden
ServiceUnavailable	503 서비스를 사용할 수 없음
SignatureDoesNotMatch	403 Forbidden
TooManyBuckets	400 잘못된 요청
UserKeyMustBeSpecified	400 잘못된 요청

StorageGRID 사용자 지정 오류 코드

이름	설명	HTTP 상태
XBucketLifecycleNotAllowed	기존 규정 준수 버킷에서는 버킷 수명 주기 구성이 허용되지 않습니다.	400 잘못된 요청
XBucketPolicyParseException	수신된 버킷 정책 JSON을 구문 분석하는 데 실패했습니다.	400 잘못된 요청
XComplianceConflict	기존 규정 준수 설정으로 인해 작업이 거부되었습니다.	403 Forbidden
XComplianceReducedRedundancyForbidden	기존 규정 준수 버킷에서는 중복성 감소가 허용되지 않습니다.	400 잘못된 요청
XMaxBucketPolicyLengthExceeded	정책이 허용된 최대 버킷 정책 길이를 초과했습니다.	400 잘못된 요청
XMissingInternalRequestHeader	내부 요청의 헤더가 누락되었습니다.	400 잘못된 요청
XNoSuchBucketCompliance	지정된 버킷에는 레거시 규정 준수 기능이 활성화되어 있지 않습니다.	404 찾을 수 없음
XNotAcceptable	요청에는 충족할 수 없는 하나 이상의 accept 헤더가 포함되어 있습니다.	406 허용되지 않음
XNotImplemented	요청하신 내용은 아직 구현되지 않은 기능을 암시합니다.	501 구현되지 않음

StorageGRID 사용자 지정 작업

StorageGRID에서 지원되는 사용자 지정 버킷 작업

StorageGRID 시스템은 S3 REST API에 추가되는 사용자 지정 작업을 지원합니다.

다음 표에는 StorageGRID에서 지원하는 사용자 지정 작업이 나열되어 있습니다.

작업	설명
"GET Bucket 일관성"	특정 버킷에 적용되는 일관성 수준을 반환합니다.
"PUT 버킷 일관성"	특정 버킷에 적용되는 일관성 수준을 설정합니다.
"GET 버킷 마지막 액세스 시간"	특정 버킷에 대해 마지막 액세스 시간 업데이트가 활성화되었는지 또는 비활성화되었는지 여부를 반환합니다.

작업	설명
"PUT Bucket 마지막 액세스 시간"	특정 버킷에 대한 마지막 액세스 시간 업데이트를 활성화 또는 비활성화할 수 있습니다.
"DELETE Bucket 메타데이터 알림 구성"	특정 버킷과 연결된 메타데이터 알림 구성 XML을 삭제합니다.
"GET 버킷 메타데이터 알림 구성"	특정 버킷과 연결된 메타데이터 알림 구성 XML을 반환합니다.
"PUT 버킷 메타데이터 알림 구성"	버킷에 대한 메타데이터 알림 서비스를 구성합니다.
"저장 공간 사용량 가져오기"	계정에서 사용 중인 총 저장 용량과 해당 계정과 연결된 각 버킷의 저장 용량을 보여줍니다.
"더 이상 사용되지 않음: CreateBucket 규정 준수 설정 포함"	더 이상 사용되지 않으며 지원되지 않습니다. 규정 준수 기능이 활성화된 새 버킷을 더 이상 생성할 수 없습니다.
"사용 중단됨: GET 버킷 규정 준수"	더 이상 사용되지 않지만 지원되는 기능: 기존 레거시 규정 준수 버킷에 현재 적용되는 규정 준수 설정을 반환합니다.
"사용 중단됨: PUT Bucket compliance"	더 이상 사용되지 않지만 지원되는 기능: 기존 레거시 규정 준수 버킷의 규정 준수 설정을 수정할 수 있습니다.

GET Bucket consistency 요청을 사용하여 **StorageGRID**에서 버킷 정합성 보장 수준을 검색합니다

GET 버킷 일관성 요청을 사용하면 특정 버킷에 적용되는 일관성 수준을 확인할 수 있습니다.

기본 일관성은 새로 생성된 객체에 대해 쓰기 후 읽기를 보장하도록 설정되어 있습니다.

이 작업을 완료하려면 s3:GetBucketConsistency 권한이 있거나 루트 계정이어야 합니다.

요청 예시

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

응답

응답 XML에서 `<Consistency>`은(는) 다음 값 중 하나를 반환합니다.

일관성	설명
모두	모든 노드가 데이터를 즉시 수신해야 하며, 그렇지 않으면 요청이 실패합니다.
strong-global	모든 사이트에서 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다.
strong-site	사이트 내 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다.
새 쓰기 후 읽기	(기본값) 새 객체에 대해서는 쓰기 후 읽기 일관성을, 객체 업데이트에 대해서는 최종 일관성을 제공합니다. 높은 가용성과 데이터 보호를 보장합니다. 대부분의 경우에 권장됩니다.
사용 가능	새로운 객체 생성 및 객체 업데이트 모두에 대해 최종 일관성을 제공합니다. S3 버킷의 경우, 필요한 경우에만 사용하십시오(예: 거의 읽지 않는 로그 값이 포함된 버킷 또는 존재하지 않는 키에 대한 HEAD 또는 GET 작업). S3 FabricPool 버킷에서는 지원되지 않습니다.

응답 예시

```
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>
```

관련 정보

"일관성"

PUT Bucket 정합성 보장 요청을 사용하여 StorageGRID에서 정합성 보장 수준 지정

PUT 버킷 일관성 요청을 사용하면 버킷에서 수행되는 작업에 적용할 일관성 수준을 지정할 수 있습니다.

기본 일관성은 새로 생성된 객체에 대해 쓰기 후 읽기를 보장하도록 설정되어 있습니다.

시작하기 전에

이 작업을 완료하려면 s3:PutBucketConsistency 권한이 있거나 루트 계정이어야 합니다.

요청

The `x-ntap-sg-consistency` 매개변수에는 다음 값 중 하나가 포함되어야 합니다.

일관성	설명
모두	모든 노드가 데이터를 즉시 수신해야 하며, 그렇지 않으면 요청이 실패합니다.
strong-global	모든 사이트에서 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다.
strong-site	사이트 내 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다.
새 쓰기 후 읽기	(기본값) 새 객체에 대해서는 쓰기 후 읽기 일관성을, 객체 업데이트에 대해서는 최종 일관성을 제공합니다. 높은 가용성과 데이터 보호를 보장합니다. 대부분의 경우에 권장됩니다.
사용 가능	새로운 객체 생성 및 객체 업데이트 모두에 대해 최종 일관성을 제공합니다. S3 버킷의 경우, 필요한 경우에만 사용하십시오(예: 거의 읽지 않는 로그 값이 포함된 버킷 또는 존재하지 않는 키에 대한 HEAD 또는 GET 작업). S3 FabricPool 버킷에서는 지원되지 않습니다.

참고: 일반적으로 "새로 쓰기 후 읽기" 일관성을 사용하는 것이 좋습니다. 요청이 제대로 작동하지 않으면 가능하면 애플리케이션 클라이언트 동작을 변경하십시오. 또는 각 API 요청에 대해 일관성을 지정하도록 클라이언트를 구성하십시오. 버킷 수준에서 일관성을 설정하는 것은 최후의 수단으로만 사용하십시오.

요청 예시

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

관련 정보

"일관성"

GET Bucket last access time 요청을 사용하여 **StorageGRID**에서 버킷 마지막 액세스 시간 상태를 검색합니다

GET 버킷 마지막 액세스 시간 요청을 사용하면 개별 버킷에 대해 마지막 액세스 시간 업데이트가 활성화되었는지 또는 비활성화되었는지 확인할 수 있습니다.

이 작업을 완료하려면 s3:GetBucketLastAccessTime 권한이 있거나 계정 루트여야 합니다.

요청 예시

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

응답 예시

이 예시는 버킷에 대해 마지막 액세스 시간 업데이트가 활성화되어 있음을 보여줍니다.

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

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

PUT Bucket last access time 요청을 사용하여 **StorageGRID**에서 마지막 액세스 시간 업데이트를 활성화 및 비활성화합니다

PUT 버킷 최종 액세스 시간 요청을 사용하면 개별 버킷에 대한 최종 액세스 시간 업데이트를 활성화 또는 비활성화할 수 있습니다. 최종 액세스 시간 업데이트를 비활성화하면 성능이 향상되며, 버전 10.3.0 이상으로 생성된 모든 버킷의 기본 설정입니다.

이 작업을 완료하려면 버킷에 대한 s3:PutBucketLastAccessTime 권한이 있거나 계정 루트여야 합니다.

> StorageGRID 버전 10.3부터 모든 새 버킷에 대해 마지막 액세스 시간 업데이트가 기본적으로 비활성화됩니다. 이전 버전의 StorageGRID를 사용하여 생성된 버킷이 있고 새 기본 동작과 일치하도록 하려면 해당 이전 버킷 각각에 대해 마지막 액세스 시간 업데이트를 명시적으로 비활성화해야 합니다. 마지막 액세스 시간 업데이트는 PUT Bucket last access time 요청을 사용하거나 테넌트 관리자의 버킷 세부 정보 페이지에서 활성화하거나 비활성화할 수 있습니다. "마지막 액세스 시간 업데이트 활성화 또는 비활성화"을 참조하십시오.

버킷에 대한 마지막 액세스 시간 업데이트가 비활성화된 경우, 해당 버킷에 대한 작업에는 다음과 같은 동작이 적용됩니다.

- GetObject, GetObjectAcl, GetObjectTagging, 및 HeadObject 요청은 마지막 액세스 시간을 업데이트하지 않습니다. 해당 객체는 정보 라이프사이클 관리(ILM) 평가를 위한 대기열에 추가되지 않습니다.
- CopyObject 및 PutObjectTagging 요청이 메타데이터만 업데이트하는 경우에도 마지막 액세스 시간이 업데이트됩니다. 해당 객체는 ILM 평가를 위한 대기열에 추가됩니다.
- 소스 버킷에 대한 마지막 액세스 시간 업데이트가 비활성화된 경우, CopyObject 요청은 소스 버킷의 마지막 액세스

시간을 업데이트하지 않습니다. 복사된 객체는 소스 버킷의 ILM 평가 대기열에 추가되지 않습니다. 그러나 대상 버킷의 경우, CopyObject 요청은 항상 마지막 액세스 시간을 업데이트합니다. 복사된 객체는 ILM 평가 대기열에 추가됩니다.

- CompleteMultipartUpload 요청은 마지막 액세스 시간을 업데이트합니다. 완료된 객체는 ILM 평가를 위한 큐에 추가됩니다.

요청 예시

이 예제는 버킷의 마지막 액세스 시간을 활성화합니다.

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

이 예제는 버킷의 마지막 액세스 시간을 비활성화합니다.

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

StorageGRID에서 DELETE 버킷 메타데이터 알림 구성 요청을 사용하여 검색 통합 서비스를 비활성화합니다

DELETE 버킷 메타데이터 알림 구성 요청을 사용하면 구성 XML을 삭제하여 개별 버킷에 대한 검색 통합 서비스를 비활성화할 수 있습니다.

이 작업을 완료하려면 버킷에 대한 s3:DeleteBucketMetadataNotification 권한이 있거나 계정 루트여야 합니다.

요청 예시

이 예시는 버킷에 대한 검색 통합 서비스를 비활성화하는 방법을 보여줍니다.

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

GET 버킷 메타데이터 알림 구성 요청을 사용하여 StorageGRID에서 검색 통합을 구성합니다

GET 버킷 메타데이터 알림 구성 요청을 사용하면 개별 버킷에 대한 검색 통합을 구성하는 데 사용되는 구성 XML을 검색할 수 있습니다.

이 작업을 완료하려면 s3:GetBucketMetadataNotification 권한이 있거나 계정 루트여야 합니다.

요청 예시

이 요청은 `bucket`라는 이름의 버킷에 대한 메타데이터 알림 구성을 검색합니다.

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

응답

응답 본문에는 버킷에 대한 메타데이터 알림 구성이 포함됩니다. 메타데이터 알림 구성을 통해 검색 통합을 위해 버킷을 구성하는 방법을 결정할 수 있습니다. 즉, 어떤 객체를 인덱싱할지, 그리고 해당 객체의 메타데이터가 어떤 엔드포인트로 전송될지를 지정할 수 있습니다.

```
<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>
```

각 메타데이터 알림 구성에는 하나 이상의 규칙이 포함됩니다. 각 규칙은 적용 대상 객체와 StorageGRID가 객체 메타데이터를 전송해야 하는 대상을 지정합니다. 대상은 StorageGRID 엔드포인트의 URN을 사용하여 지정해야 합니다.

이름	설명	필수
MetadataNotificationConfiguration	메타데이터 알림의 객체와 대상을 지정하는 데 사용되는 규칙의 컨테이너 태그입니다. 하나 이상의 Rule 요소를 포함합니다.	예

이름	설명	필수
규칙	지정된 인덱스에 메타데이터를 추가해야 하는 객체를 식별하는 규칙에 대한 컨테이너 태그입니다. 접두사가 겹치는 규칙은 허용되지 않습니다. MetadataNotificationConfiguration 요소에 포함되어 있습니다.	예
ID	규칙에 대한 고유 식별자입니다. 규칙 요소에 포함됩니다.	아니요
상태	상태는 '활성화됨' 또는 '비활성화됨'일 수 있습니다. 비활성화된 규칙에 대해서는 아무런 조치도 취해지지 않습니다. 규칙 요소에 포함됩니다.	예
접두사	접두사와 일치하는 객체는 해당 규칙의 영향을 받으며, 해당 객체의 메타데이터가 지정된 대상으로 전송됩니다. 모든 객체와 일치시키려면 빈 접두사를 지정하십시오. 규칙 요소에 포함됩니다.	예
목적지	규칙의 대상에 대한 컨테이너 태그입니다. 규칙 요소에 포함됩니다.	예

이름	설명	필수
Urn	객체 메타데이터가 전송되는 대상의 URN입니다. 다음 속성을 가진 StorageGRID 엔드포인트의 URN이어야 합니다. • `es`세 번째 요소여야 합니다. • URN은 메타데이터가 저장된 인덱스와 유형으로 끝나야 하며, 형식은 `domain-name/myindex/mytype`입니다. 엔드포인트는 테넌트 관리자 또는 테넌트 관리 API를 사용하여 구성됩니다. 다음과 같은 형식을 사용합니다. • `arn:aws:es:_region:account-ID_:domain/mydomain/myindex/mytype` • `urn:mysite:es:::mydomain/myindex/mytype` 엔드포인트는 구성 XML을 제출하기 전에 구성되어야 합니다. 그렇지 않으면 구성이 실패하고 404 오류가 발생합니다. URN은 대상 요소에 포함됩니다.	예

응답 예시

`<MetadataNotificationConfiguration></MetadataNotificationConfiguration>`태그 사이에 포함된 XML은 버킷에 대한 검색 통합 엔드포인트와의 통합 구성 방식을 보여줍니다. 이 예에서는 객체 메타데이터가 `current`라는 이름의 Elasticsearch 인덱스와 `2017`라는 유형으로 전송되며, 이 인덱스는 `records`라는 이름의 AWS 도메인에 호스팅됩니다.

```
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>
```

관련 정보

"테넌트 계정 사용"

PUT Bucket 메타데이터 알림 구성 요청을 사용하여 StorageGRID에서 검색 통합 서비스를 활성화합니다

PUT 버킷 메타데이터 알림 구성 요청을 사용하면 개별 버킷에 대해 검색 통합 서비스를 활성화할 수 있습니다. 요청 본문에 제공하는 메타데이터 알림 구성 XML은 메타데이터가 대상 검색 인덱스로 전송되는 객체를 지정합니다.

이 작업을 완료하려면 버킷에 대한 s3:PutBucketMetadataNotification 권한이 있거나 계정 루트여야 합니다.

요청

요청 본문에는 메타데이터 알림 구성이 포함되어야 합니다. 각 메타데이터 알림 구성에는 하나 이상의 규칙이 포함됩니다. 각 규칙은 적용되는 객체와 StorageGRID가 객체 메타데이터를 전송해야 하는 대상을 지정합니다.

객체는 객체 이름의 접두사를 기준으로 필터링할 수 있습니다. 예를 들어, 접두사가 `/images`인 객체의 메타데이터는 한 대상으로 보내고, 접두사가 `/videos`인 객체는 다른 대상으로 보낼 수 있습니다.

접두사가 겹치는 구성은 유효하지 않으며 제출 시 거부됩니다. 예를 들어, 접두사가 `test`인 객체에 대한 규칙 하나와 접두사가 `test2`인 객체에 대한 규칙 하나를 포함하는 구성은 허용되지 않습니다.

대상은 StorageGRID 엔드포인트의 URN을 사용하여 지정해야 합니다. 메타데이터 알림 구성이 제출될 때 엔드포인트가 존재해야 하며, 그렇지 않으면 요청이 실패합니다 `400 Bad Request`. 오류 메시지는 다음과 같습니다. `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>
```

이 표는 메타데이터 알림 구성 XML의 요소를 설명합니다.

이름	설명	필수
MetadataNotificationConfiguration	메타데이터 알림의 객체와 대상을 지정하는 데 사용되는 규칙의 컨테이너 태그입니다. 하나 이상의 Rule 요소를 포함합니다.	예
규칙	지정된 인덱스에 메타데이터를 추가해야 하는 객체를 식별하는 규칙에 대한 컨테이너 태그입니다. 접두사가 겹치는 규칙은 허용되지 않습니다. MetadataNotificationConfiguration 요소에 포함되어 있습니다.	예
ID	규칙에 대한 고유 식별자입니다. 규칙 요소에 포함됩니다.	아니요
상태	상태는 '활성화됨' 또는 '비활성화됨'일 수 있습니다. 비활성화된 규칙에 대해서는 아무런 조치도 취해지지 않습니다. 규칙 요소에 포함됩니다.	예

이름	설명	필수
접두사	접두사와 일치하는 객체는 해당 규칙의 영향을 받으며, 해당 객체의 메타데이터가 지정된 대상으로 전송됩니다. 모든 객체와 일치시키려면 빈 접두사를 지정하십시오. 규칙 요소에 포함됩니다.	예
목적지	규칙의 대상에 대한 컨테이너 태그입니다. 규칙 요소에 포함됩니다.	예
Urn	객체 메타데이터가 전송되는 대상의 URN입니다. 다음 속성을 가진 StorageGRID 엔드포인트의 URN이어야 합니다. • `es`세 번째 요소여야 합니다. • URN은 메타데이터가 저장된 인덱스와 유형으로 끝나야 하며, 형식은 `domain-name/myindex/mytype`입니다. 엔드포인트는 테넌트 관리자 또는 테넌트 관리 API를 사용하여 구성됩니다. 다음과 같은 형식을 사용합니다. • `arn:aws:es:region:account-ID:domain/mydomain/myindex/mytype` • `urn:mysite:es:::mydomain/myindex/mytype` 엔드포인트는 구성 XML을 제출하기 전에 구성되어야 합니다. 그렇지 않으면 구성이 실패하고 404 오류가 발생합니다. URN은 대상 요소에 포함됩니다.	예

요청 예시

이 예시는 버킷에 대한 검색 통합을 활성화하는 방법을 보여줍니다. 이 예시에서는 모든 객체의 메타데이터가 동일한 대상으로 전송됩니다.

```
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>
```

이 예시에서는 접두사 `/images`와 일치하는 객체의 객체 메타데이터는 한 대상으로 전송되고, 접두사 `/videos`와 일치하는 객체의 객체 메타데이터는 두 번째 대상으로 전송됩니다.

```
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>
```

검색 통합 서비스에서 생성된 **JSON**

버킷에 대한 검색 통합 서비스를 활성화하면 객체 메타데이터 또는 태그가 추가, 업데이트 또는 삭제될 때마다 JSON 문서가 생성되어 대상 엔드포인트로 전송됩니다.

이 예시는 키가 `SGWS/Tagging.txt`인 객체가 `test`라는 이름의 버킷에 생성될 때 생성될 수 있는 `JSON`의 예시를 보여줍니다. 해당 `test` 버킷은 버전 관리가 되지 않으므로 `versionId` 태그는 비어 있습니다.

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

메타데이터 알림에 포함된 객체 메타데이터

이 표는 검색 통합이 활성화되었을 때 대상 엔드포인트로 전송되는 JSON 문서에 포함된 모든 필드를 나열합니다.

문서 이름에는 버킷 이름, 객체 이름, 그리고 있는 경우 버전 ID가 포함됩니다.

유형	품목명	설명
버킷 및 객체 정보	버킷	버킷 이름
버킷 및 객체 정보	키	객체 키 이름
버킷 및 객체 정보	versionID	버전 관리 버킷에 있는 객체의 객체 버전입니다.
버킷 및 객체 정보	지역	버킷 영역(예: `us-east-1`
시스템 메타데이터	크기	HTTP 클라이언트에 표시되는 객체 크기(바이트)
시스템 메타데이터	md5	객체 해시

유형	품목명	설명
사용자 메타데이터	메타데이터 *key:value*	객체에 대한 모든 사용자 메타데이터(키-값 쌍)
태그	태그 *key:value*	객체에 대해 정의된 모든 객체 태그(키-값 쌍)

태그 및 사용자 메타데이터의 경우, StorageGRID는 날짜와 숫자를 문자열 또는 S3 이벤트 알림으로 Elasticsearch에 전달합니다. Elasticsearch가 이러한 문자열을 날짜 또는 숫자로 해석하도록 구성하려면 동적 필드 매핑 및 날짜 형식 매핑에 대한 Elasticsearch 지침을 따르세요. 검색 통합 서비스를 구성하기 전에 인덱스에서 동적 필드 매핑을 활성화해야 합니다. 문서가 인덱싱된 후에는 인덱스에서 문서의 필드 유형을 편집할 수 없습니다.

관련 정보

"테넌트 계정 사용"

GET Storage Usage 요청을 사용하여 StorageGRID에서 계정 및 버킷 스토리지 사용량을 조회합니다

GET 스토리지 사용량 요청은 계정에서 사용 중인 총 스토리지 용량과 해당 계정과 연결된 각 버킷의 스토리지 용량을 알려줍니다.

계정 및 버킷별 저장 용량은 `x-ntap-sg-usage` 쿼리 매개변수를 사용하여 수정된 ListBuckets 요청을 통해 확인할 수 있습니다. 버킷 저장 용량 사용량은 시스템에서 처리하는 PUT 및 DELETE 요청과 별도로 추적됩니다. 특히 시스템 부하가 높은 경우, 요청 처리 속도에 따라 사용량 값이 예상 값과 일치하기까지 다소 시간이 걸릴 수 있습니다.

기본적으로 StorageGRID는 강력한 전역 일관성을 사용하여 사용량 정보를 검색하려고 시도합니다. 강력한 전역 일관성을 달성할 수 없는 경우 StorageGRID는 강력한 사이트 일관성을 사용하여 사용량 정보를 검색하려고 시도합니다.

이 작업을 완료하려면 s3:ListAllMyBuckets 권한이 있거나 루트 계정이어야 합니다.

요청 예시

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

응답 예시

이 예시는 두 개의 버킷에 네 개의 객체와 12바이트의 데이터가 있는 계정을 보여줍니다. 각 버킷에는 두 개의 객체와 6바이트의 데이터가 포함되어 있습니다.

```
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>
```

버전 관리

저장된 모든 객체 버전은 응답의 `ObjectCount` 및 `DataBytes` 값에 영향을 미칩니다. 삭제 마커는 `ObjectCount` 총합에 추가되지 않습니다.

관련 정보

"일관성"

기존 규정 준수를 위한 더 이상 사용되지 않는 버킷 요청

기존 **StorageGRID** 규정 준수를 위한 더 이상 사용되지 않는 버킷 요청

StorageGRID 기존 규정 준수 기능을 사용하여 생성된 버킷을 관리하려면 StorageGRID S3 REST API를 사용해야 할 수도 있습니다.

규정 준수 기능이 더 이상 사용되지 않습니다.

이전 StorageGRID 버전에서 제공되었던 StorageGRID 규정 준수 기능은 더 이상 사용되지 않으며 S3 Object Lock으로 대체되었습니다.

이전에 전역 규정 준수 설정을 활성화한 경우, StorageGRID 11.6에서는 전역 S3 객체 잠금 설정이 활성화됩니다. 규정 준수가 활성화된 새 버킷은 더 이상 생성할 수 없습니다. 하지만 필요한 경우 StorageGRID S3 REST API를 사용하여 기존의 레거시 규정 준수 버킷을 관리할 수 있습니다.

- "S3 REST API를 사용하여 S3 Object Lock 구성"
- "ILM을 사용하여 객체 관리"
- "NetApp 기술 자료: StorageGRID 11.5에서 기존 규정 준수 버킷을 관리하는 방법"

더 이상 사용되지 않는 규정 준수 요청:

- "사용 중단됨 - 규정 준수를 위한 PUT 버킷 요청 수정"

  SGCompliance XML 요소는 더 이상 사용되지 않습니다. 이전에는 PUT Bucket 요청의 선택적 XML 요청 본문에 이 StorageGRID 사용자 지정 요소를 포함하여 규정 준수 버킷을 생성할 수 있었습니다.

- "사용 중단됨 - GET 버킷 규정 준수"

  GET 버킷 규정 준수 요청은 더 이상 사용되지 않습니다. 하지만 기존 레거시 규정 준수 버킷에 현재 적용 중인 규정 준수 설정을 확인하려면 이 요청을 계속 사용할 수 있습니다.

- "사용 중단됨 - PUT Bucket compliance"

  PUT 버킷 규정 준수 요청은 더 이상 사용되지 않습니다. 하지만 기존의 레거시 규정 준수 버킷에 대한 규정 준수 설정을 수정하는 데에는 이 요청을 계속 사용할 수 있습니다. 예를 들어 기존 버킷에 법적 보존 조치를 적용하거나 보존 기간을 늘릴 수 있습니다.

StorageGRID의 더 이상 사용되지 않는 **SGCompliance** 요소

SGCompliance XML 요소는 더 이상 사용되지 않습니다. 이전에는 CreateBucket 요청의 선택적 XML 요청 본문에 이 StorageGRID 사용자 지정 요소를 포함하여 규정 준수 버킷을 생성할 수 있었습니다.

> 이전 StorageGRID 버전에서 제공되었던 StorageGRID 규정 준수 기능은 더 이상 사용되지 않으며 S3 Object Lock으로 대체되었습니다. 자세한 내용은 다음을 참조하십시오.
>
> - "S3 REST API를 사용하여 S3 Object Lock 구성"
> - "NetApp 기술 자료: StorageGRID 11.5에서 기존 규정 준수 버킷을 관리하는 방법"

규정 준수 기능이 활성화된 새 버킷을 더 이상 생성할 수 없습니다. 규정 준수를 위한 CreateBucket 요청 수정을 사용하여 새 규정 준수 버킷을 생성하려고 하면 다음 오류 메시지가 반환됩니다.

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

StorageGRID의 더 이상 사용되지 않는 GET 버킷 규정 준수 요청

GET 버킷 규정 준수 요청은 더 이상 사용되지 않습니다. 하지만 기존 레거시 규정 준수 버킷에 현재 적용 중인 규정 준수 설정을 확인하려면 이 요청을 계속 사용할 수 있습니다.

> 이전 StorageGRID 버전에서 제공되었던 StorageGRID 규정 준수 기능은 더 이상 사용되지 않으며 S3 Object Lock으로 대체되었습니다. 자세한 내용은 다음을 참조하십시오.
>
> - "S3 REST API를 사용하여 S3 Object Lock 구성"
> - "NetApp 기술 자료: StorageGRID 11.5에서 기존 규정 준수 버킷을 관리하는 방법"

이 작업을 완료하려면 s3:GetBucketCompliance 권한이 있거나 루트 계정이어야 합니다.

요청 예시

이 예시 요청을 통해 이름이 `mybucket`인 버킷에 대한 규정 준수 설정을 확인할 수 있습니다.

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

응답 예시

응답 XML에는 `<SGCompliance>`버킷에 적용되는 규정 준수 설정이 나열됩니다. 이 예시 응답은 각 객체가 그리드에 수집된 시점부터 1년(525,600분) 동안 보존되는 버킷의 규정 준수 설정을 보여줍니다. 현재 이 버킷에는 법적 보존 조치가 적용되지 않았습니다. 각 객체는 1년 후 자동으로 삭제됩니다.

```
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>
```

이름	설명
RetentionPeriodMinutes	이 버킷에 추가된 객체의 보존 기간(분)입니다. 보존 기간은 객체가 그리드에 수집된 시점부터 시작됩니다.
LegalHold	• 참: 이 버킷은 현재 법적 보존 조치가 적용되어 있습니다. 이 버킷의 객체는 보존 기간이 만료되었더라도 법적 보존 조치가 해제될 때까지 삭제할 수 없습니다. • 거짓: 이 버킷은 현재 법적 보존 상태가 아닙니다. 이 버킷의 객체는 보존 기간이 만료되면 삭제될 수 있습니다.
AutoDelete	• True: 이 버킷에 있는 객체는 보존 기간이 만료되면 자동으로 삭제됩니다. 단, 버킷이 법적 보존 상태에 있는 경우는 예외입니다. • 거짓: 이 버킷의 객체는 보존 기간이 만료되어도 자동으로 삭제되지 않습니다. 삭제해야 하는 경우 수동으로 삭제해야 합니다.

오류 응답

버킷이 규정을 준수하도록 생성되지 않은 경우 응답의 HTTP 상태 코드는 `404 Not Found`이며 S3 오류 코드는 `XNoSuchBucketCompliance`입니다.

StorageGRID의 더 이상 사용되지 않는 **PUT** 버킷 규정 준수 요청

PUT 버킷 규정 준수 요청은 더 이상 사용되지 않습니다. 하지만 기존의 레거시 규정 준수 버킷에 대한 규정 준수 설정을 수정하는 데에는 이 요청을 계속 사용할 수 있습니다. 예를 들어 기존 버킷에 법적 보존 조치를 적용하거나 보존 기간을 늘릴 수 있습니다.

> 이전 StorageGRID 버전에서 제공되었던 StorageGRID 규정 준수 기능은 더 이상 사용되지 않으며 S3 Object Lock으로 대체되었습니다. 자세한 내용은 다음을 참조하십시오.
>
> • "S3 REST API를 사용하여 S3 Object Lock 구성"
> • "NetApp 기술 자료: StorageGRID 11.5에서 기존 규정 준수 버킷을 관리하는 방법"

이 작업을 완료하려면 s3:PutBucketCompliance 권한이 있거나 루트 계정이어야 합니다.

PUT 버킷 규정 준수 요청을 발행할 때는 규정 준수 설정의 모든 필드에 값을 지정해야 합니다.

요청 예시

이 예시 요청은 `mybucket`라는 이름의 버킷에 대한 규정 준수 설정을 수정합니다. 이 예시에서는 `mybucket` 버킷에 저장된 객체가 그리드에 수집된 시점부터 1년이 아닌 2년(1,051,200분) 동안 보존됩니다. 이 버킷에는 법적 보존이 설정되어 있지 않습니다. 각 객체는 2년 후 자동으로 삭제됩니다.

```
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>
```

이름	설명
RetentionPeriodMinutes	이 버킷에 추가된 객체의 보존 기간(분)입니다. 보존 기간은 객체가 그리드에 수집된 시점부터 시작됩니다. 중요 RetentionPeriodMinutes에 새 값을 지정할 때는 버킷의 현재 보존 기간보다 크거나 같은 값을 지정해야 합니다. 버킷의 보존 기간이 설정되면 해당 값을 줄일 수 없으며, 늘릴 수만 있습니다.
LegalHold	• 참: 이 버킷은 현재 법적 보존 조치가 적용되어 있습니다. 이 버킷의 객체는 보존 기간이 만료되었더라도 법적 보존 조치가 해제될 때까지 삭제할 수 없습니다. • 거짓: 이 버킷은 현재 법적 보존 상태가 아닙니다. 이 버킷의 객체는 보존 기간이 만료되면 삭제될 수 있습니다.
AutoDelete	• True: 이 버킷에 있는 객체는 보존 기간이 만료되면 자동으로 삭제됩니다. 단, 버킷이 법적 보존 상태에 있는 경우는 예외입니다. • 거짓: 이 버킷의 객체는 보존 기간이 만료되어도 자동으로 삭제되지 않습니다. 삭제해야 하는 경우 수동으로 삭제해야 합니다.

규정 준수 설정의 일관성

PUT 버킷 규정 준수 요청을 사용하여 S3 버킷의 규정 준수 설정을 업데이트하면 StorageGRID는 그리드 전체에서 버킷의 메타데이터를 업데이트하려고 시도합니다. 기본적으로 StorageGRID는 **Strong-global** 일관성을 사용하여 모든 데이터 센터 사이트와 버킷 메타데이터를 포함하는 모든 스토리지 노드가 변경된 규정 준수 설정에 대해 쓰기 후 읽기 일관성을 갖도록 보장합니다.

데이터 센터 사이트 또는 해당 사이트의 여러 스토리지 노드를 사용할 수 없어 StorageGRID가 **Strong-global** 일관성을 달성할 수 없는 경우, 응답에 대한 HTTP 상태 코드는 `503 Service Unavailable.`

이 응답을 받으면 그리드 관리자에게 문의하여 필요한 스토리지 서비스를 최대한 빨리 사용할 수 있도록 해야 합니다. 그리드 관리자가 각 사이트에서 충분한 스토리지 노드를 사용할 수 있도록 확보할 수 없는 경우, 기술 지원 팀에서 **Strong-site** 일관성을 강제로 적용하여 실패한 요청을 다시 시도하도록 안내할 수 있습니다.

기술 지원 팀의 지시가 있거나 해당 수준 사용의 잠재적 결과를 완전히 이해하지 않는 한, PUT 버킷 규정 준수를 위해 **Strong-site** 일관성을 강제로 적용하지 마십시오.

일관성 수준을 *강력한 사이트(Strong-site)*로 낮추면 StorageGRID는 업데이트된 규정 준수 설정이 해당 사이트 내의 클라이언트 요청에 대해서만 쓰기 후 읽기 일관성을 유지하도록 보장합니다. 즉, 모든 사이트와 스토리지 노드를 사용할 수 있을 때까지 StorageGRID 시스템은 해당 버킷에 대해 일시적으로 여러 개의 일관되지 않은 설정을 유지할 수 있습니다. 이러한 일관되지 않은 설정은 예기치 않고 바람직하지 않은 동작을 초래할 수 있습니다. 예를 들어, 버킷을 법적 보존 상태로 설정한 후 일관성 수준을 낮추면 버킷의 이전 규정 준수 설정(즉, 법적 보존 해제)이 일부 데이터 센터 사이트에서 계속 적용될 수 있습니다. 결과적으로, 법적 보존 상태라고 생각했던 객체가 보존 기간이 만료되면 사용자 또는 AutoDelete(활성화된 경우)에 의해 삭제될 수 있습니다.

Strong-site 일관성 사용을 강제하려면 PUT Bucket 규정 준수 요청을 다시 실행하고 다음과 같이 `Consistency-Control` HTTP 요청 헤더를 포함하십시오.

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

오류 응답

- 버킷이 규정을 준수하도록 생성되지 않은 경우 응답의 HTTP 상태 코드는 `404 Not Found`입니다.
- 요청의 `RetentionPeriodMinutes`이 버킷의 현재 보존 기간보다 짧은 경우 HTTP 상태 코드는 `400 Bad Request`입니다.

관련 정보

"사용 중단 예정: 규정 준수를 위한 PUT 버킷 요청 수정"

${post_edited_translations.segment}

StorageGRID에서 **AWS** 정책을 사용하여 **S3** 버킷 및 오브젝트에 대한 액세스를 제어하는 방법

StorageGRID는 Amazon Web Services(AWS) 정책 언어를 사용하여 S3 테넌트가 버킷 및 해당 버킷 내의 객체에 대한 액세스를 제어할 수 있도록 합니다. StorageGRID 시스템은 S3 REST API 정책 언어의 일부를 구현합니다. S3 API에 대한 액세스 정책은 JSON 형식으로 작성됩니다.

액세스 정책 개요

StorageGRID는 세 가지 유형의 액세스 정책을 지원합니다.

- *버킷 정책*은 GetBucketPolicy, PutBucketPolicy, DeleteBucketPolicy S3 API 작업 또는 Tenant Manager 또는 Tenant Management API를 사용하여 관리됩니다. 버킷 정책은 버킷에 연결되어 버킷 소유자 계정 또는 다른 계정의 사용자가 해당 버킷 및 버킷 내 객체에 액세스하는 것을 제어하도록 구성됩니다. 버킷 정책은 하나의 버킷과 여러 그룹에만 적용됩니다.
- *그룹 정책*은 테넌트 관리자 또는 테넌트 관리 API를 사용하여 구성됩니다. 그룹 정책은 계정의 그룹에 연결되어 해당 그룹이 계정 소유의 특정 리소스에 액세스할 수 있도록 허용합니다. 그룹 정책은 하나의 그룹과 여러 버킷에만 적용됩니다.

- 세션 정책: AssumeRole 요청을 생성할 때 포함됩니다. 세션 정책은 해당 세션에만 적용되며, 그룹 및 버킷 정책에서 부여된 권한 외에 사용자가 갖는 권한을 추가로 정의합니다.

> 그룹, 버킷 및 세션 정책 간에는 우선순위 차이가 없습니다.

StorageGRID 버킷 및 그룹 정책은 Amazon에서 정의한 특정 문법을 따릅니다. 각 정책 내부에는 정책 문 배열이 있으며, 각 문에는 다음 요소가 포함됩니다.

- Statement ID(Sid)(선택 사항)
- 효과
- Principal/NotPrincipal
- 리소스/NotResource
- Action/NotAction
- 조건 (선택 사항)

정책 문은 다음과 같은 구조를 사용하여 권한을 지정하도록 구성됩니다. <Condition>이(가) 적용될 때 <Principal>이(가) <Resource>에서 <Action>을(를) 수행하는 것을 허용/거부하기 위해 <Effect>을(를) 부여합니다.

각 정책 요소는 특정 기능을 위해 사용됩니다.

요소	설명
Sid	Sid 요소는 선택 사항입니다. Sid는 사용자에 대한 설명 용도로만 사용됩니다. Sid는 저장되지만 StorageGRID 시스템에서 해석되지는 않습니다.
효과	Effect 요소를 사용하여 지정된 작업이 허용되는지 또는 거부되는지 여부를 설정합니다. 지원되는 Action 요소 키워드를 사용하여 버킷 또는 객체에 대해 허용(또는 거부)할 작업을 식별해야 합니다.
Principal/NotPrincipal	사용자, 그룹 및 계정이 특정 리소스에 액세스하고 특정 작업을 수행하도록 허용할 수 있습니다. 요청에 S3 서명이 포함되어 있지 않은 경우, 보안 주체로 와일드카드 문자(*)를 지정하여 익명 액세스가 허용됩니다. 기본적으로 계정 루트만 해당 계정이 소유한 리소스에 액세스할 수 있습니다. 버킷 정책에서는 Principal 요소만 지정하면 됩니다. 그룹 정책의 경우, 정책이 연결된 그룹이 암묵적인 Principal 요소가 됩니다.
리소스/NotResource	Resource 요소는 버킷과 객체를 식별합니다. Amazon 리소스 이름(ARN)을 사용하여 리소스를 식별하고 버킷 및 객체에 대한 권한을 허용하거나 거부할 수 있습니다.
Action/NotAction	권한에는 작업(Action)과 결과(Effect)라는 두 가지 구성 요소가 있습니다. 그룹이 리소스를 요청하면 해당 리소스에 대한 액세스 권한이 부여되거나 거부됩니다. 명시적으로 권한을 할당하지 않으면 액세스가 거부되지만, 명시적 거부를 사용하여 다른 정책에서 부여한 권한을 재정의할 수 있습니다.

요소	설명
상태	조건 요소는 선택 사항입니다. 조건을 사용하면 정책을 적용해야 하는 시기를 결정하는 표현식을 만들 수 있습니다.

Action 요소에서는 와일드카드 문자(*)를 사용하여 모든 작업 또는 작업의 하위 집합을 지정할 수 있습니다. 예를 들어, 이 Action은 s3:GetObject, s3:PutObject, s3:DeleteObject 등의 권한과 일치합니다.

```
s3:*Object
```

리소스 요소에서는 와일드카드 문자(*)와 (?)를 사용할 수 있습니다. 별표(*)는 0개 이상의 문자와 일치하고, 물음표(?)는 임의의 단일 문자와 일치합니다.

Principal 요소에서는 익명 액세스(모든 사용자에게 권한을 부여)를 설정하는 경우를 제외하고는 와일드카드 문자가 지원되지 않습니다. 예를 들어 Principal 값으로 와일드카드(*)를 설정할 수 있습니다.

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

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

다음 예시에서는 Effect, Principal, Action 및 Resource 요소를 사용하는 정책 문을 보여줍니다. 이 예시는 "Allow"라는 Effect를 사용하여 Principal, admin 그룹 `federated-group/admin` 및 finance 그룹 `federated-group/finance`에 `s3:ListBucket`이라는 이름의 버킷에 대한 Action `mybucket`과 해당 버킷 내의 모든 객체에 대한 Action `s3:GetObject`을 수행할 수 있는 권한을 부여하는 전체 버킷 정책 문을 보여줍니다.

```
{
  "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/*"
      ]
    }
  ]
}
```

버킷 정책의 크기 제한은 20,480바이트이고, 그룹 정책의 크기 제한은 5,120바이트입니다.

정책의 일관성

기본적으로 그룹 정책에 대한 모든 업데이트는 최종 일관성을 유지합니다. 그룹 정책 일관성을 갖게 되면 정책 캐싱으로 인해 변경 사항이 적용되는 데 최대 15분이 추가로 소요될 수 있습니다. 기본적으로 버킷 정책에 대한 모든 업데이트는 강력 일관성을 유지합니다.

필요에 따라 버킷 정책 업데이트에 대한 일관성 보장을 변경할 수 있습니다. 예를 들어, 사이트 장애 발생 시에도 버킷 정책 변경 사항을 적용할 수 있도록 설정할 수 있습니다.

이 경우 `Consistency-Control` 헤더를 PutBucketPolicy 요청에 설정하거나 PUT 버킷 일관성 요청을 사용할 수 있습니다. 버킷 정책이 일관성을 확보하면 정책 캐싱으로 인해 변경 사항이 적용되는 데 최대 8초가 추가로 소요될 수 있습니다.

> 일시적인 상황을 해결하기 위해 일관성 값을 다른 값으로 설정한 경우, 작업이 완료되면 버킷 수준 설정을 원래 값으로 되돌려야 합니다. 그렇지 않으면 향후 모든 버킷 요청에 수정된 설정이 적용됩니다.

세션 정책이란 무엇인가요?

세션 정책은 사용자가 그룹을 수임할 때와 같이 특정 세션 동안 사용할 수 있는 권한을 일시적으로 제한하는 액세스 정책입니다. 세션 정책은 권한의 하위 집합만 허용할 수 있으며, 추가 권한을 부여할 수는 없습니다. 그룹 자체는 더 광범위한 권한을 가질 수 있습니다.

${post_edited_translations.segment}

정책 설명에서 ARN은 Principal 요소와 Resource 요소에 사용됩니다.

- ${post_edited_translations.segment}

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

- 다음 구문을 사용하여 ID 리소스 ARN(사용자 및 그룹)을 지정합니다.

```
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
```

기타 고려 사항:

- ${post_edited_translations.segment}
- 객체 키에 지정할 수 있는 국제 문자는 JSON UTF-8 또는 JSON \u 이스케이프 시퀀스를 사용하여 인코딩해야 합니다. 퍼센트 인코딩은 지원되지 않습니다.

  "RFC 2141 URN 구문"

  PutBucketPolicy 작업에 대한 HTTP 요청 본문은 charset=UTF-8로 인코딩되어야 합니다.

정책에 리소스 지정

정책 문에서 Resource 요소를 사용하여 권한이 허용되거나 거부될 버킷 또는 객체를 지정할 수 있습니다.

- 각 정책 설명에는 리소스 요소가 필요합니다. 정책에서 리소스는 요소 `Resource`로 표시되며, 제외의 경우 `NotResource`로 표시됩니다.
- S3 리소스 ARN을 사용하여 리소스를 지정합니다. 예를 들면 다음과 같습니다.

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

- 객체 키 내부에 정책 변수를 사용할 수도 있습니다. 예를 들면 다음과 같습니다.

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

- 리소스 값은 그룹 정책을 생성할 때 아직 존재하지 않는 버킷을 지정할 수 있습니다.

정책에서 보안 주체 지정

Principal 요소를 사용하여 정책 문에 따라 리소스에 대한 액세스가 허용/거부된 사용자, 그룹 또는 테넌트 계정을 식별합니다.

- 버킷 정책의 각 정책 설명에는 Principal 요소가 포함되어야 합니다. 그룹 정책의 정책 설명은 그룹 자체가 Principal로 간주되므로 Principal 요소가 필요하지 않습니다.
- 정책에서 주체는 "Principal" 요소로 표시되며, 제외의 경우에는 "NotPrincipal"로 표시됩니다.
- 계정 기반 ID는 ID 또는 ARN을 사용하여 지정해야 합니다.

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

- 이 예제에서는 테넌트 계정 루트와 해당 계정에 속한 모든 사용자를 포함하는 테넌트 계정 ID 27233906934684427525를 사용합니다.

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

- 계정 루트만 지정할 수도 있습니다.

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

- 특정 연합 사용자("Alex")를 지정할 수 있습니다.

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

- 특정 페더레이션 그룹("Managers")을 지정할 수 있습니다.

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

- 익명 주체를 지정할 수 있습니다.

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

- 혼동을 피하려면 사용자 이름 대신 사용자 UUID를 사용할 수 있습니다.

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

예를 들어, 알렉스가 조직을 떠나고 해당 사용자 이름 `Alex`이 삭제되었다고 가정해 보겠습니다. 새로운 알렉스가 조직에 합류하여 동일한 `Alex` 사용자 이름을 할당받으면, 새 사용자는 의도치 않게 원래 사용자에게 부여된 권한을 상속받을 수 있습니다.

- 주요 값은 버킷 정책을 생성할 때 아직 존재하지 않는 그룹/사용자 이름을 지정할 수 있습니다.

정책에서 권한 지정

정책에서 Action 요소는 리소스에 대한 권한을 허용/거부하는 데 사용됩니다. 정책에서 지정할 수 있는 권한 집합은 "Action" 요소 또는 제외를 나타내는 "NotAction" 요소로 표시됩니다. 이러한 각 요소는 특정 S3 REST API 작업에 매핑됩니다.

해당 표에는 버킷에 적용되는 권한과 객체에 적용되는 권한이 나열되어 있습니다.

Amazon S3는 이제 PutBucketReplication 및 DeleteBucketReplication 작업 모두에 s3:PutReplicationConfiguration 권한을 사용합니다. StorageGRID는 각 작업에 대해 별도의 권한을 사용하며, 이는 원래 Amazon S3 사양과 일치합니다.

삭제는 put 작업을 사용하여 기존 값을 덮어쓸 때 수행됩니다.

버킷에 적용되는 권한

권한	**S3 REST API** 작업	**StorageGRID**에 대한 사용자 지정
s3:CreateBucket	CreateBucket	예. 참고: 그룹 정책에서만 사용합니다.
s3:DeleteBucket	DeleteBucket	
s3:DeleteBucketMetadataNotification	DELETE Bucket 메타데이터 알림 구성	예
s3:DeleteBucketPolicy	DeleteBucketPolicy	
s3:DeleteReplicationConfiguration	DeleteBucketReplication	예, PUT 및 DELETE에 대한 별도의 권한
s3:GetBucketAcl	GetBucketAcl	
s3:GetBucketCompliance	GET Bucket compliance(지원 중단됨)	예

권한	**S3 REST API** 작업	**StorageGRID**에 대한 사용자 지정
s3:GetBucketConsistency	GET Bucket 일관성	예
s3:GetBucketCORS	GetBucketCors	
s3:GetEncryptionConfiguration	GetBucketEncryption	
s3:GetBucketLastAccessTime	GET 버킷 마지막 액세스 시간	예
s3:GetBucketLocation	GetBucketLocation	
s3:GetBucketMetadataNotification	GET 버킷 메타데이터 알림 구성	예
s3:GetBucketNotification	GetBucketNotificationConfiguration	
s3:GetBucketObjectLockConfiguration	GetObjectLockConfiguration	
s3:GetBucketPolicy	GetBucketPolicy	
s3:GetBucketTagging	GetBucketTagging	
s3:GetBucketVersioning	GetBucketVersioning	
s3:GetLifecycleConfiguration	GetBucketLifecycleConfiguration	
s3:GetReplicationConfiguration	GetBucketReplication	
s3:ListAllMyBuckets	• ListBuckets • 저장 공간 사용량 가져오기	예, GET Storage Usage의 경우 그렇습니다. 참고: 그룹 정책에서만 사용합니다.
s3:ListBucket	• ListObjects • HeadBucket • RestoreObject	
s3:ListBucketMultipartUploads	• ListMultipartUploads • RestoreObject	
s3:ListBucketVersions	GET Bucket 버전	

권한	**S3 REST API** 작업	**StorageGRID**에 대한 사용자 지정
s3:PutBucketCompliance	PUT Bucket compliance(지원 중단됨)	예
s3:PutBucketConsistency	PUT 버킷 일관성	예
s3:PutBucketCORS	• DeleteBucketCors† • PutBucketCors	
s3:PutEncryptionConfiguration	• DeleteBucketEncryption • PutBucketEncryption	
s3:PutBucketLastAccessTime	PUT Bucket 마지막 액세스 시간	예
s3:PutBucketMetadataNotification	PUT 버킷 메타데이터 알림 구성	예
s3:PutBucketNotification	PutBucketNotificationConfiguration	
s3:PutBucketObjectLockConfiguration	• CreateBucket with the `x-amz-bucket-object-lock-enabled: true` 요청 헤더(s3:CreateBucket 권한도 필요) • PutObjectLockConfiguration	
s3:PutBucketPolicy	PutBucketPolicy	
s3:PutBucketTagging	• DeleteBucketTagging† • PutBucketTagging	
s3:PutBucketVersioning	PutBucketVersioning	
s3:PutLifecycleConfiguration	• DeleteBucketLifecycle† • PutBucketLifecycleConfiguration	
s3:PutReplicationConfiguration	PutBucketReplication	예, PUT 및 DELETE에 대한 별도의 권한

객체에 적용되는 권한

권한	**S3 REST API** 작업	**StorageGRID**에 대한 사용자 지정
s3:AbortMultipartUpload	• AbortMultipartUpload • RestoreObject	
s3:BypassGovernanceRetention	• DeleteObject • DeleteObjects • PutObjectRetention	
s3:DeleteObject	• DeleteObject • DeleteObjects • RestoreObject	
s3:DeleteObjectTagging	DeleteObjectTagging	
s3:DeleteObjectVersionTagging	DeleteObjectTagging(객체의 특정 버전)	
s3:DeleteObjectVersion	DeleteObject(특정 버전의 객체)	
s3:GetObject	• GetObject • HeadObject • RestoreObject • SelectObjectContent	
s3:GetObjectAcl	GetObjectAcl	
s3:GetObjectLegalHold	GetObjectLegalHold	
s3:GetObjectRetention	GetObjectRetention	
s3:GetObjectTagging	GetObjectTagging	
s3:GetObjectVersionTagging	GetObjectTagging(객체의 특정 버전)	
s3:GetObjectVersion	GetObject(객체의 특정 버전)	
s3:ListMultipartUploadParts	ListParts, RestoreObject	

권한	**S3 REST API** 작업	**StorageGRID**에 대한 사용자 지정
s3:PutObject	• PutObject • CopyObject • RestoreObject • CreateMultipartUpload • CompleteMultipartUpload • UploadPart • UploadPartCopy	
s3:PutObjectLegalHold	PutObjectLegalHold	
s3:PutObjectRetention	PutObjectRetention	
s3:PutObjectTagging	PutObjectTagging	
s3:PutObjectVersionTagging	PutObjectTagging(객체의 특정 버전)	
s3:PutOverwriteObject	• PutObject • CopyObject • PutObjectTagging • DeleteObjectTagging • CompleteMultipartUpload	예
s3:RestoreObject	RestoreObject	

PutOverwriteObject 권한 사용

s3:PutOverwriteObject 권한은 객체를 생성하거나 업데이트하는 작업에 적용되는 사용자 지정 StorageGRID 권한입니다. 이 권한 설정에 따라 클라이언트가 객체의 데이터, 사용자 정의 메타데이터 또는 S3 객체 태깅을 덮어쓸 수 있는지 여부가 결정됩니다.

이 권한에 대한 가능한 설정은 다음과 같습니다.

- 허용: 클라이언트가 객체를 덮어쓸 수 있습니다. 이것이 기본 설정입니다.
- 거부: 클라이언트는 객체를 덮어쓸 수 없습니다. '거부'로 설정하면 PutOverwriteObject 권한은 다음과 같이 작동합니다.
  - 동일한 경로에서 기존 객체가 발견된 경우:
    - 객체의 데이터, 사용자 정의 메타데이터 또는 S3 객체 태깅은 덮어쓸 수 없습니다.
    - 진행 중인 모든 데이터 수집 작업이 취소되고 오류가 반환됩니다.
    - S3 버전 관리가 활성화된 경우, '거부' 설정은 PutObjectTagging 또는 DeleteObjectTagging 작업이 객체 및 해당 객체의 현재 버전이 아닌 버전의 TagSet을 수정하는 것을 방지합니다.

◦ 기존 개체를 찾을 수 없는 경우 이 권한은 아무런 효력이 없습니다.

- 이 권한이 없는 경우, 허용 권한이 설정된 것과 동일한 효과가 나타납니다.

> 현재 S3 정책에서 덮어쓰기를 허용하고 PutOverwriteObject 권한이 거부로 설정되어 있는 경우, 클라이언트는 객체의 데이터, 사용자 정의 메타데이터 또는 객체 태그를 덮어쓸 수 없습니다. 또한, 클라이언트 수정 방지 확인란이 선택된 경우(구성 > 보안 설정 > 네트워크 및 객체), 해당 설정이 PutOverwriteObject 권한 설정을 재정의합니다.

정책에 조건 지정

조건은 정책이 적용되는 시점을 정의합니다. 조건은 연산자와 키-값 쌍으로 구성됩니다.

조건은 평가를 위해 키-값 쌍을 사용합니다. 조건 요소는 여러 조건을 포함할 수 있으며, 각 조건은 여러 키-값 쌍을 포함할 수 있습니다. 조건 블록은 다음 형식을 사용합니다.

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

다음 예에서 IpAddress 조건은 SourceIp 조건 키를 사용합니다.

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

지원되는 조건 연산자

조건 연산자는 다음과 같이 분류됩니다.

- 문자열
- 숫자
- 부울
- IP 주소
- 널 검사

조건 연산자	설명
StringEquals	키와 문자열 값을 정확한 일치(대소문자 구분)를 기준으로 비교합니다.
StringNotEquals	키와 문자열 값을 대소문자를 구분하는 부정 일치 방식을 기반으로 비교합니다.

조건 연산자	설명
StringEqualsIgnoreCase	키와 문자열 값을 정확히 일치하는지 여부에 따라 비교합니다(대소문자 구분 없음).
StringNotEqualsIgnoreCase	키와 문자열 값을 비교할 때 대소문자를 구분하지 않고 부정 일치 방식을 사용합니다.
StringLike	키와 문자열 값을 정확히 일치 여부(대소문자 구분)로 비교합니다. * 및 ? 와일드카드 문자를 사용할 수 있습니다.
StringNotLike	키와 문자열 값을 대소문자를 구분하는 부정 일치 방식으로 비교합니다. * 및 ? 와일드카드 문자를 사용할 수 있습니다.
NumericEquals	정확한 일치 여부를 기준으로 키와 숫자 값을 비교합니다.
NumericNotEquals	키와 숫자 값을 부정 일치 방식을 기반으로 비교합니다.
NumericGreaterThan	"보다 큼"을 기준으로 키와 숫자 값을 비교합니다.
NumericGreaterThanEquals	"크거나 같음"을 기준으로 키와 숫자 값을 비교합니다.
NumericLessThan	"보다 작음"을 기준으로 키와 숫자 값을 비교합니다.
NumericLessThanEquals	"작거나 같음"을 기준으로 키와 숫자 값을 비교합니다.
부울	키를 부울 값과 비교하여 "참 또는 거짓" 일치 여부를 확인합니다.
IpAddress	키를 IP 주소 또는 IP 주소 범위와 비교합니다.
NotIpAddress	키를 IP 주소 또는 IP 주소 범위와 부정 일치 방식으로 비교합니다.
Null	현재 요청 컨텍스트에 조건 키가 있는지 확인합니다.
IfExists	Null 조건을 제외한 모든 조건 연산자에 추가하여 해당 조건 키의 부재 여부를 확인합니다. 조건 키가 없으면 TRUE를 반환합니다.

지원되는 조건 키

조건 키	작업	설명
aws:SourceIp	IP 연산자	요청이 전송된 IP 주소와 비교합니다. 버킷 또는 객체 작업에 사용할 수 있습니다. 참고**:** S3 요청이 관리 노드 및 게이트웨이 노드의 로드 밸런서 서비스를 통해 전송된 경우, 이는 로드 밸런서 서비스 업스트림의 IP 주소와 비교됩니다. 참고: 타사의 비투명 로드 밸런서를 사용하는 경우, 해당 로드 밸런서의 IP 주소와 비교합니다. 유효성을 확인할 수 없으므로 모든 `X-Forwarded-For` 헤더는 무시됩니다.
aws:username	리소스/식별	요청을 보낸 발신자의 사용자 이름과 비교합니다. 버킷 또는 객체 작업에 사용할 수 있습니다.
s3:구분 기호	s3:ListBucket 및 s3:ListBucketVersions 권한	구분 기호 매개변수는 ListObjects 또는 ListObjectVersions 요청에 지정된 구분 기호 매개변수와 비교합니다.
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	기존 객체에 특정 태그 키와 값이 있어야 합니다.
s3:max-keys	s3:ListBucket 및 s3:ListBucketVersions 권한	ListObjects 또는 ListObjectVersions 요청에 지정된 max-keys 매개변수와 비교합니다.

조건 키	작업	설명
s3:object-lock-mode	s3:PutObject	요청 헤더에서 확장된 `object-lock-mode`을(를) PutObject, CopyObject 및 CreateMultipartUpload 요청과 비교합니다.
s3:object-lock-mode	s3:PutObjectRetention	`object-lock-mode`PutObjectRetention 요청의 XML 본문에서 확장된 내용과 비교합니다.
s3:object-lock-remaining-retention-days	s3:PutObject	`x-amz-object-lock-retain-until-date` 요청 헤더에 지정된 보존 기한 또는 버킷 기본 보존 기간에서 계산된 값과 비교하여 이러한 값이 다음 요청에 대해 허용 가능한 범위 내에 있는지 확인합니다. • PutObject • CopyObject • CreateMultipartUpload
s3:object-lock-remaining-retention-days	s3:PutObjectRetention	PutObjectRetention 요청에 지정된 보존 기한과 비교하여 허용 범위 내에 있는지 확인합니다.
s3:prefix	s3:ListBucket 및 s3:ListBucketVersions 권한	ListObjects 또는 ListObjectVersions 요청에 지정된 접두사 매개변수와 비교합니다.
s3:RequestObjectTag/<tag-key>	s3:PutObject s3:PutObjectTagging s3:PutObjectVersionTagging	객체 요청에 태깅이 포함된 경우 특정 태그 키와 값이 필요합니다.
s3:x-amz-server-side-encryption-customer-algorithm	s3:PutObject	요청 헤더에서 확장된 `sse-customer-algorithm` 또는 `copy-source-sse-customer-algorithm`와 비교합니다. PutObject, CopyObject, CreateMultipartUpload, UploadPart, UploadPartCopy, CompleteMultipartUpload 요청의 요청 헤더에서 확장된 내용과 비교합니다.

정책에 변수 지정

정책에서 변수를 사용하여 정책 정보가 있을 때 이를 채울 수 있습니다. 정책 변수는 `Resource` 요소 내에서 또는

`Condition` 요소 내의 문자열 비교에서 사용할 수 있습니다.

이 예시에서 변수 `${aws:username}`는 Resource 요소의 일부입니다.

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

이 예시에서 변수 `${aws:username}`는 조건 블록의 조건 값의 일부입니다.

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

변수	설명
`${aws:SourceIp}`	SourceIp 키를 제공된 변수로 사용합니다.
`${aws:username}`	제공된 변수로 사용자 이름 키를 사용합니다.
`${s3:prefix}`	제공된 변수로 서비스별 접두사 키를 사용합니다.
`${s3:max-keys}`	제공된 변수로 서비스별 max-keys 키를 사용합니다.
`${*}`	특수 문자입니다. 해당 문자를 문자 그대로 사용합니다.
`${?}`	특수 문자입니다. 해당 문자를 문자 그대로 물음표(?)로 사용합니다.
`${$}`	특수 문자입니다. 해당 문자를 문자 그대로 '$'로 사용합니다.

특별 처리가 필요한 정책 생성

정책에 따라 보안이나 지속적인 운영에 위험을 초래할 수 있는 권한이 부여될 수 있습니다. 예를 들어 계정의 루트 사용자를 차단하는 경우가 있습니다. StorageGRID S3 REST API 구현은 Amazon보다 정책 유효성 검사 시 덜 엄격하지만, 정책 평가 시에는 동일하게 엄격합니다.

정책 설명	정책 유형	**Amazon** 동작	**StorageGRID** 동작
루트 계정에 대한 모든 권한을 스스로 거부합니다.	버킷	유효하고 시행 중이지만, 루트 사용자 계정은 모든 S3 버킷 정책 작업에 대한 권한을 유지합니다.	동일

정책 설명	정책 유형	**Amazon** 동작	**StorageGRID** 동작
사용자/그룹에 대한 모든 권한을 자신에게 거부	그룹	유효하고 강제력이 있습니다	동일
외부 계정 그룹에 모든 권한 허용	버킷	잘못된 주체	유효하지만 정책에서 허용된 경우에도 모든 S3 버킷 정책 작업에 대한 권한이 405 Method Not Allowed 오류를 반환합니다
외부 계정 루트 또는 사용자에게 모든 권한 허용	버킷	유효하지만 정책에서 허용된 경우에도 모든 S3 버킷 정책 작업에 대한 권한이 405 Method Not Allowed 오류를 반환합니다	동일
모든 사용자에게 모든 작업에 대한 권한을 부여합니다.	버킷	유효하지만, 모든 S3 버킷 정책 작업에 대한 권한이 외부 계정 루트 사용자 및 사용자에 대해 405 Method Not Allowed 오류를 반환합니다.	동일
모든 사용자에게 모든 작업에 대한 권한을 거부합니다.	버킷	유효하고 시행 중이지만, 루트 사용자 계정은 모든 S3 버킷 정책 작업에 대한 권한을 유지합니다.	동일
Principal은 존재하지 않는 사용자 또는 그룹입니다.	버킷	잘못된 주체	유효한
리소스는 존재하지 않는 S3 버킷입니다.	그룹	유효한	동일
Principal은 로컬 그룹입니다.	버킷	잘못된 주체	유효한
정책은 소유자가 아닌 계정(익명 계정 포함)에 개체를 배치할 수 있는 권한을 부여합니다.	버킷	유효합니다. 객체는 생성자 계정의 소유이며 버킷 정책이 적용되지 않습니다. 생성자 계정은 객체 ACL을 사용하여 객체에 대한 액세스 권한을 부여해야 합니다.	유효합니다. 객체는 버킷 소유자 계정의 소유입니다. 버킷 정책이 적용됩니다.

한 번만 쓰고 여러 번 읽을 수 있는**(WORM)** 보호

데이터, 사용자 정의 객체 메타데이터 및 S3 객체 태깅을 보호하기 위해 WORM(Write-Once-Read-Many) 버킷을 생성할 수 있습니다. WORM 버킷은 새 객체 생성을 허용하고 기존 콘텐츠의 덮어쓰기 또는 삭제를 방지하도록 구성합니다. 여기에 설명된 방법 중 하나를 사용하십시오.

덮어쓰기가 항상 거부되도록 하려면 다음을 수행할 수 있습니다.

- 그리드 관리자에서 구성 > 보안 > 보안 설정 > 네트워크 및 객체*로 이동하여 *클라이언트 수정 방지 확인란을 선택합니다.
- 다음 규칙 및 S3 정책을 적용하십시오.
  - S3 정책에 PutOverwriteObject DENY 작업을 추가합니다.
  - DeleteObject DENY 작업을 S3 정책에 추가합니다.
  - S3 정책에 PutObject ALLOW 작업을 추가합니다.

> S3 정책에서 DeleteObject를 DENY로 설정하더라도 "30일 후 복사본 0개"와 같은 규칙이 있는 경우 ILM이 객체를 삭제하는 것을 막을 수는 없습니다.

> 이러한 모든 규칙과 정책을 적용하더라도 동시 쓰기를 방지할 수는 없습니다(상황 A 참조). 하지만 순차적으로 완료된 덮어쓰기는 방지할 수 있습니다(상황 B 참조).

상황 **A**: 동시 쓰기(방지되지 않음)

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

상황 **B**: 순차적으로 완료된 덮어쓰기(방지됨)

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

관련 정보

- "StorageGRID ILM 규칙이 객체를 관리하는 방법"
- "예시 버킷 정책"
- "예시 그룹 정책"
- "예시 세션 정책"
- "ILM을 사용하여 객체 관리"
- "테넌트 계정 사용"

StorageGRID S3 REST API 세션 정책 예

다음 예제를 사용하여 StorageGRID 세션 정책을 구성하십시오.

예시: 객체 검색을 허용하는 세션 정책 설정

이 예시에서 세션의 주체는 bucket1에서만 객체를 검색할 수 있습니다. "s3:PutOverwriteObject" 권한 사용과 같은 StorageGRID 관련 작업을 제외한 모든 다른 작업은 암묵적으로 거부됩니다. 세션 정책은 AssumeRole API를 호출할

때 JSON 파일로 제공할 수 있습니다.

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

StorageGRID S3 REST API 버킷 정책 예

이 섹션의 예제를 사용하여 버킷에 대한 StorageGRID 액세스 정책을 구성하십시오.

버킷 정책은 해당 정책이 적용된 버킷에 대한 액세스 권한을 지정합니다. S3 PutBucketPolicy API를 사용하여 다음 도구 중 하나로 버킷 정책을 구성할 수 있습니다.

- "테넌트 관리자".
- AWS CLI에서 다음 명령을 사용하는 방법(참조 "버킷에 대한 작업"):

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

예시: 모든 사용자에게 버킷에 대한 읽기 전용 액세스 권한 허용

이 예시에서는 익명 사용자를 포함한 모든 사용자가 버킷의 객체 목록을 조회하고 버킷의 모든 객체에 대해 GetObject 작업을 수행할 수 있습니다. 그 외의 모든 작업은 거부됩니다. 단, 계정 루트 사용자를 제외한 다른 사용자는 버킷에 쓰기 권한이 없으므로 이 정책은 그다지 유용하지 않을 수 있습니다.

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

예시: 한 계정의 모든 사용자에게 버킷에 대한 전체 액세스 권한을 부여하고, 다른 계정의 모든 사용자에게는 읽기 전용 액세스 권한만 부여합니다.

이 예시에서는 지정된 한 계정의 모든 사용자에게 버킷에 대한 전체 액세스 권한이 부여되는 반면, 다른 지정된 계정의 모든 사용자에게는 버킷 목록을 조회하고 `shared/` 객체 키 접두사로 시작하는 버킷 내 객체에 대한 GetObject 작업만 수행할 수 있도록 허용됩니다.

> StorageGRID에서 소유자가 아닌 계정(익명 계정 포함)이 생성한 객체는 버킷 소유자 계정의 소유가 됩니다. 버킷 정책은 이러한 객체에 적용됩니다.

```
{
  "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/*"
        }
      }
    }
  ]
}
```

예시: 모든 사용자에게 버킷에 대한 읽기 전용 액세스 권한을 부여하고 지정된 그룹에게는 전체 액세스 권한을 부여합니다.

이 예시에서는 익명 사용자를 포함한 모든 사용자가 버킷의 목록을 조회하고 GetObject 작업을 버킷 내 모든 객체에 대해 수행할 수 있지만, 지정된 계정의 그룹 `Marketing`에 속한 사용자에게만 전체 액세스 권한이 부여됩니다.

```
{
  "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/*"
      ]
    }
  ]
}
```

예시: 클라이언트의 **IP** 주소 범위 내에 있는 모든 사용자에게 버킷에 대한 읽기 및 쓰기 권한을 부여합니다.

이 예시에서는 익명 사용자를 포함한 모든 사용자가 지정된 IP 범위(54.240.143.0~54.240.143.255, 단 54.240.143.188 제외)에서 요청이 들어온 경우 버킷 목록을 조회하고 버킷 내 모든 객체에 대한 모든 객체 작업을 수행할 수 있습니다. 다른 모든 작업은 거부되며, 지정된 IP 범위를 벗어난 모든 요청은 차단됩니다.

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

예시: 지정된 페더레이션 사용자에게만 버킷에 대한 전체 액세스 권한 허용

이 예시에서 연합 사용자 Alex는 `examplebucket`버킷과 그 객체에 대한 모든 접근 권한을 갖습니다. ''root'를 포함한 다른 모든 사용자는 모든 작업이 명시적으로 거부됩니다. 단, ''root'는 Put/Get/DeleteBucketPolicy에 대한 권한이 거부되지 않습니다.

```
{
  "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/*"
      ]
    }
  ]
}
```

예시: **PutOverwriteObject** 권한

이 예에서 `Deny` PutOverwriteObject 및 DeleteObject에 대한 Effect는 누구도 객체의 데이터, 사용자 정의 메타데이터 및 S3 객체 태깅을 덮어쓰거나 삭제할 수 없도록 합니다.

```
{
  "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/*"
    }
  ]
}
```

StorageGRID S3 REST API 그룹 정책 예

이 섹션의 예제를 사용하여 그룹에 대한 StorageGRID 액세스 정책을 구성하십시오.

그룹 정책은 해당 정책이 적용된 그룹에 대한 액세스 권한을 지정합니다. 정책에는 `Principal` 요소가 없는데, 이는 암묵적으로 포함되어 있기 때문입니다. 그룹 정책은 Tenant Manager 또는 API를 사용하여 구성합니다.

예시: **Tenant Manager**를 사용하여 그룹 정책 설정

테넌트 관리자에서 그룹을 추가하거나 편집할 때 그룹 정책을 선택하여 해당 그룹 구성원이 가질 S3 액세스 권한을 결정할 수 있습니다. "S3 테넌트에 대한 그룹 생성"을 참조하십시오.

- **S3** 액세스 없음: 기본 옵션입니다. 이 그룹의 사용자는 버킷 정책을 통해 액세스 권한이 부여되지 않는 한 S3 리소스에 액세스할 수 없습니다. 이 옵션을 선택하면 기본적으로 루트 사용자만 S3 리소스에 액세스할 수 있습니다.
- 읽기 전용 액세스: 이 그룹의 사용자는 S3 리소스에 대한 읽기 전용 액세스 권한을 갖습니다. 예를 들어, 이 그룹의 사용자는 객체 목록을 보고 객체 데이터, 메타데이터 및 태그를 읽을 수 있습니다. 이 옵션을 선택하면 읽기 전용 그룹 정책에 대한 JSON 문자열이 텍스트 상자에 나타납니다. 이 문자열은 편집할 수 없습니다.
- 전체 액세스: 이 그룹의 사용자는 버킷을 포함한 S3 리소스에 대한 전체 액세스 권한을 갖습니다. 이 옵션을 선택하면 전체 액세스 그룹 정책에 대한 JSON 문자열이 텍스트 상자에 나타납니다. 이 문자열은 편집할 수 없습니다.
- 랜섬웨어 완화: 이 샘플 정책은 이 테넌트의 모든 버킷에 적용됩니다. 이 그룹의 사용자는 일반적인 작업을 수행할 수 있지만, 객체 버전 관리가 활성화된 버킷에서 객체를 영구적으로 삭제할 수는 없습니다.

  '모든 버킷 관리' 권한이 있는 Tenant Manager 사용자는 이 그룹 정책을 재정의할 수 있습니다. '모든 버킷 관리' 권한은 신뢰할 수 있는 사용자에게만 부여하고, 가능한 경우 다단계 인증(MFA)을 사용하십시오.

- 사용자 지정: 그룹의 사용자에게는 텍스트 상자에 지정한 권한이 부여됩니다.

예시: 모든 버킷에 대한 그룹의 전체 액세스 권한 부여

이 예시에서는 버킷 정책에서 명시적으로 거부하지 않는 한, 그룹의 모든 구성원은 테넌트 계정이 소유한 모든 버킷에 대한 전체 액세스 권한을 갖습니다.

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

예시: 모든 버킷에 대해 그룹별 읽기 전용 액세스 권한 허용

이 예시에서, 버킷 정책에 의해 명시적으로 거부되지 않는 한, 그룹의 모든 구성원은 S3 리소스에 대한 읽기 전용 액세스 권한을 갖습니다. 예를 들어, 이 그룹의 사용자는 객체 목록을 조회하고 객체 데이터, 메타데이터 및 태그를 읽을 수 있습니다.

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

예시: 그룹 구성원에게 버킷 내 자신의 "폴더"에 대한 전체 액세스 권한만 부여합니다.

이 예시에서 그룹 구성원은 지정된 버킷 내에서 자신의 특정 폴더(키 접두사)만 나열하고 접근할 수 있습니다. 이러한 폴더의 개인 정보 보호 수준을 결정할 때는 다른 그룹 정책 및 버킷 정책의 접근 권한도 고려해야 합니다.

```
{
  "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}/*"
    }
  ]
}
```

StorageGRID 감사 로그에 기록된 S3 작업

감사 메시지는 StorageGRID 서비스에서 생성되어 텍스트 로그 파일에 저장됩니다. 감사 로그에서 S3 관련 감사 메시지를 검토하여 버킷 및 객체 작업에 대한 자세한 정보를 얻을 수 있습니다.

감사 로그에서 추적되는 버킷 작업

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

감사 로그에 기록된 객체 작업

- CompleteMultipartUpload
- CopyObject
- DeleteObject
- GetObject
- HeadObject
- PutObject
- RestoreObject
- SelectObject
- UploadPart (ILM 규칙에서 균형 또는 엄격 수집을 사용하는 경우)
- UploadPartCopy(ILM 규칙에서 균형 또는 엄격 수집을 사용하는 경우)

관련 정보

- "감사 로그 파일 액세스"
- "클라이언트 쓰기 감사 메시지"
- "클라이언트 읽기 감사 메시지"