S3 REST API 사용

StorageGRID 11.8

NetApp
March 19, 2024

목차

S3 REST API 사용 . . . 1
- S3 REST API 지원 버전 및 업데이트 . . . 1
- 빠른 참조: 지원되는 S3 API 요청 . . . 3
- S3 REST API 구성을 테스트합니다 . . . 22
- StorageGRID에서 S3 REST API를 구현하는 방법 . . . 24
- Amazon S3 REST API 지원 . . . 38
- StorageGRID 사용자 정의 작업 . . . 83
- 버킷 및 그룹 액세스 정책 . . . 103
- 감사 로그에서 S3 작업을 추적했습니다 . . . 128

S3 REST API 사용

S3 REST API 지원 버전 및 업데이트

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

S3 REST API를 지원하므로 StorageGRID 시스템을 사용하는 사내 오브젝트 스토리지를 통해 S3 웹 서비스용으로 개발된 서비스 지향 애플리케이션을 연결할 수 있습니다. 클라이언트 애플리케이션의 현재 S3 REST API 호출 사용에 대한 최소 변경 사항이 필요합니다.

지원되는 버전

StorageGRID는 다음과 같은 S3 및 HTTP 버전을 지원합니다.

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

S3 REST API 지원 업데이트

놓습니다	설명
11.8	에 사용된 이름과 일치하도록 S3 작업의 이름을 업데이트했습니다 "AWS(Amazon Web Services) 문서: Amazon Simple Storage Service API Reference 를 참조하십시오".
11.7	• 추가되었습니다 "빠른 참조: 지원되는 S3 API 요청". • S3 Object Lock을 통한 거버넌스 모드 사용을 지원합니다. • StorageGRID에 대한 지원이 추가되었습니다 `x-ntap-sg-cgr-replication-status` Get Object 및 Head Object 요청에 대한 응답 헤더입니다. 이 헤더는 크로스 그리드 복제를 위한 객체의 복제 상태를 제공합니다. • SelectObjectContent 요청이 이제 Parquet 객체를 지원합니다.

놓습니다	설명
11.6	• 에 대한 지원이 추가되었습니다 `partNumber` Get Object 및 Head Object 요청의 요청 매개 변수입니다. • S3 오브젝트 잠금의 버킷 레벨에서 기본 보존 모드 및 기본 보존 기간에 대한 지원이 추가되었습니다. • 에 대한 지원이 추가되었습니다 `s3:object-lock-remaining-retention-days` Policy Condition 키 - 객체에 허용되는 보존 기간 범위를 설정합니다. • 단일 PUT 객체 작업의 Maximum_Recommended_size를 5GiB(5,368,709,120바이트)로 변경했습니다. 5GiB보다 큰 객체가 있는 경우 대신 멀티파트 업로드를 사용합니다.
11.5	• 버킷 암호화 관리에 대한 지원이 추가되었습니다. • S3 오브젝트 잠금 및 더 이상 사용되지 않는 레거시 규정 준수 요청에 대한 지원 추가 • 버전이 있는 버킷에서 여러 오브젝트 삭제 사용에 대한 지원이 추가되었습니다. • 를 클릭합니다 `Content-MD5` 이제 요청 헤더가 올바르게 지원됩니다.
11.4	• 버킷 태그 삭제, 버킷 태그 지정 가져오기 및 버킷 태그 지정을 위한 지원이 추가되었습니다. 비용 할당 태그는 지원되지 않습니다. • StorageGRID 11.4에서 만든 버킷의 경우 성능 모범 사례에 맞게 개체 키 이름을 제한하는 것이 더 이상 필요하지 않습니다. • 에서 버킷 알림에 대한 지원이 추가되었습니다 `s3:ObjectRestore:Post` 이벤트 유형입니다. • 이제 여러 파트에 대한 AWS 크기 제한이 적용됩니다. 멀티파트 업로드의 각 파트는 5MiB에서 5GiB 사이여야 합니다. 마지막 부분은 5MiB보다 작을 수 있습니다. • TLS 1.3에 대한 지원이 추가되었습니다
11.3	• 고객이 제공한 키(SSE-C)를 사용하여 오브젝트 데이터의 서버측 암호화에 대한 지원이 추가되었습니다. • 버킷 수명주기 작업(만료 작업에만 해당) 및 에 대한 삭제, 가져오기 및 Put 지원 추가 `x-amz-expiration` 응답 헤더. • 수집 시 동기식 배치를 사용하는 ILM 규칙의 영향을 설명하기 위해 PUT 개체, Put Object-Copy 및 MultiPart Upload가 업데이트되었습니다. • TLS 1.1 암호가 더 이상 지원되지 않습니다.
11.2	클라우드 스토리지 풀과 함께 사용할 POST 오브젝트 복원에 대한 지원이 추가되었습니다. 그룹 및 버킷 정책에서 ARN, 정책 조건 키 및 정책 변수에 대해 AWS 구문 사용을 지원합니다. StorageGRID 구문을 사용하는 기존 그룹 및 버킷 정책은 계속 지원됩니다. • 참고: * 사용자 지정 StorageGRID 기능에 사용되는 것을 포함하여 다른 구성 JSON/XML에서 ARN/URN을 사용하는 것은 변경되지 않았습니다.
11.1	CORS(Cross-Origin Resource Sharing), 그리드 노드에 대한 S3 클라이언트 연결을 위한 HTTP 및 버킷에 대한 규정 준수 설정에 대한 지원이 추가되었습니다.

놓습니다	설명
11.0	버킷에 대한 플랫폼 서비스(CloudMirror 복제, 알림 및 Elasticsearch 검색 통합) 구성 지원 추가 또한 버킷에 대한 객체 태그 위치 제약 조건 및 사용 가능한 정합성 보장에 대한 지원이 추가되었습니다.
10.4	버전 관리, 끝점 도메인 이름 페이지 업데이트, 정책, 정책 예제 및 PutOverwriteObject 권한에 대한 ILM 검색 변경 사항에 대한 지원이 추가되었습니다.
10.3	버전 관리 지원 추가.
10.2	그룹 및 버킷 액세스 정책 및 다중 파트 복제본(업로드 부분 복사)에 대한 지원이 추가되었습니다.
10.1	멀티파트 업로드, 가상 호스팅 스타일 요청 및 v4 인증에 대한 지원이 추가되었습니다.
10.0	StorageGRID 시스템에서 S3 REST API의 초기 지원. 현재 지원되는 _Simple Storage Service API Reference_는 2006-03-01입니다.

빠른 참조: 지원되는 **S3 API** 요청

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

이 페이지에는 StorageGRID에서 지원하는 S3 작업만 포함됩니다.

각 작업에 대한 AWS 설명서를 보려면 제목에서 링크를 선택합니다.

공통 **URI** 쿼리 매개 변수 및 요청 헤더

별도로 지정하지 않으면 다음과 같은 공통 URI 쿼리 매개 변수가 지원됩니다.

- `versionId` (오브젝트 작업에 필요한 경우)

별도로 명시되지 않은 경우 다음과 같은 일반적인 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: 일반 요청 헤더"

"AbortMultipartUpload 를 클릭합니다"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청과 함께 다음과 같은 추가 URI 쿼리 매개 변수가 추가되었습니다.

- `uploadId`

요청 본문

없음

StorageGRID 설명서

"멀티파트 업로드 작업"

"CompleteMultipartUpload를 클릭합니다"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청과 함께 다음과 같은 추가 URI 쿼리 매개 변수가 추가되었습니다.

- `uploadId`

본문 **XML** 태그를 요청합니다

StorageGRID는 다음과 같은 요청 본문 XML 태그를 지원합니다.

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

StorageGRID 설명서

"CompleteMultipartUpload를 클릭합니다"

"CopyObject 를 선택합니다"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청과 함께 다음과 같은 추가 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-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

없음

StorageGRID 설명서

"CopyObject 를 선택합니다"

"CreateBucket"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청과 함께 다음과 같은 추가 헤더가 있습니다.

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

요청 본문

StorageGRID는 구현 시 Amazon S3 REST API에 의해 정의된 모든 요청 본문 매개 변수를 지원합니다.

StorageGRID 설명서

"버킷 작업"

"CreateMultptUpload 를 클릭합니다"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청과 함께 다음과 같은 추가 헤더가 있습니다.

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

없음

StorageGRID 설명서

"CreateMultptUpload 를 클릭합니다"

"삭제 버킷"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

StorageGRID 설명서

"버킷 작업"

"DeleteBucketCors"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"버킷 작업"

"DeleteBucketEncryption"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"버킷 작업"

"DeleteBucketLifecycle"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

- "버킷 작업"
- "S3 라이프사이클 구성을 생성합니다"

"DeleteBuckketPolicy를 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"버킷 작업"

"DeleteBuckReplication 을 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"버킷 작업"

"삭제 BucketTagging"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"버킷 작업"

"DeleteObject 를 클릭합니다"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청과 함께 다음과 같은 추가 요청 헤더가 추가되었습니다.

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

요청 본문

없음

StorageGRID 설명서

"객체에 대한 작업"

"DeleteObjects 를 클릭합니다"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청과 함께 다음과 같은 추가 요청 헤더가 추가되었습니다.

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

요청 본문

StorageGRID는 구현 시 Amazon S3 REST API에 의해 정의된 모든 요청 본문 매개 변수를 지원합니다.

StorageGRID 설명서

"객체에 대한 작업"

"DeleteObjectTagging 을 선택합니다"

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"객체에 대한 작업"

"GetBuckketAcl"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"버킷 작업"

"GetBuckketCors 를 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"버킷 작업"

"GetBuckketEncryption을 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"버킷 작업"

"GetBuckLifecycleConfiguration 을 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

- "버킷 작업"
- "S3 라이프사이클 구성을 생성합니다"

"GetBucketLocation 을 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"버킷 작업"

"GetBuckNotificationConfiguration 을 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"버킷 작업"

"GetBuckketPolicy를 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"버킷 작업"

"GetBucketReplication 을 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"버킷 작업"

"GetBucketTagging"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"버킷 작업"

"GetBuckketVersioning 을 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"버킷 작업"

"GetObject 를 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청과 함께 다음과 같은 추가 URI 쿼리 매개 변수가 추가되었습니다.

- `partNumber`
- `response-cache-control`
- `response-content-disposition`
- `response-content-encoding`
- `response-content-language`
- `response-content-type`
- `response-expires`

그리고 이러한 추가 요청 헤더는 다음과 같습니다.

- `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

없음

StorageGRID 설명서

"GetObject 를 참조하십시오"

"GetObjectAcl"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"객체에 대한 작업"

"GetObjectLegalHold 를 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다"

"GetObjectLockConfiguration 을 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다"

"GetObjectRetention을 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다"

"GetObjectTagging"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"객체에 대한 작업"

"머리버킷"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

없음

StorageGRID 설명서

"버킷 작업"

"HeadObject 를 선택합니다"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청과 함께 다음과 같은 추가 헤더가 있습니다.

- `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
없음

StorageGRID 설명서
"HeadObject 를 선택합니다"

"ListBucket"

Uri 쿼리 매개 변수 및 요청 헤더
StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문
없음

StorageGRID 설명서
"서비스 및 GT, ListBucket에 대한 작업"

"ListMultipartUploads 를 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더
StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청에 다음 추가 매개 변수를 추가합니다.

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

요청 본문
없음

StorageGRID 설명서
"ListMultipartUploads 를 참조하십시오"

"ListObjects 를 선택합니다"

Uri 쿼리 매개 변수 및 요청 헤더
StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청에 다음 추가 매개 변수를 추가합니다.

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

- `max-keys`
- `prefix`

요청 본문
없음

StorageGRID 설명서
"버킷 작업"

"ListObjectsV2 를 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더
StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청에 다음 추가 매개 변수를 추가합니다.

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

요청 본문
없음

StorageGRID 설명서
"버킷 작업"

"ListObjectVersions 를 선택합니다"

Uri 쿼리 매개 변수 및 요청 헤더
StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청에 다음 추가 매개 변수를 추가합니다.

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

요청 본문
없음

StorageGRID 설명서

"버킷 작업"

"목록 파트"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청에 다음 추가 매개 변수를 추가합니다.

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

요청 본문

없음

StorageGRID 설명서

"ListMultipartUploads 를 참조하십시오"

"BuckketCors의"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

StorageGRID는 구현 시 Amazon S3 REST API에 의해 정의된 모든 요청 본문 매개 변수를 지원합니다.

StorageGRID 설명서

"버킷 작업"

"PutBucketEncryption을 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

본문 **XML** 태그를 요청합니다

StorageGRID는 다음과 같은 요청 본문 XML 태그를 지원합니다.

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

StorageGRID 설명서

"버킷 작업"

"PutBucketLifecycleConfiguration을 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

본문 **XML** 태그를 요청합니다

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 설명서

- "버킷 작업"
- "S3 라이프사이클 구성을 생성합니다"

"PutBucketNotificationConfiguration을 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

본문 **XML** 태그를 요청합니다

StorageGRID는 다음과 같은 요청 본문 XML 태그를 지원합니다.

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

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

StorageGRID 설명서

"버킷 작업"

"BucketPolicy를 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

지원되는 JSON 본문 필드에 대한 자세한 내용은 을 참조하십시오 "버킷 및 그룹 액세스 정책을 사용합니다".

"PutBucketReplication을 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

본문 **XML** 태그를 요청합니다

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

StorageGRID 설명서

"버킷 작업"

"BucketTagging"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

StorageGRID는 구현 시 Amazon S3 REST API에 의해 정의된 모든 요청 본문 매개 변수를 지원합니다.

StorageGRID 설명서

"버킷 작업"

"PutBucketVersioning을 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

바디 매개 변수를 요청합니다

StorageGRID는 다음과 같은 요청 본문 매개 변수를 지원합니다.

- `VersioningConfiguration`
- `Status`

StorageGRID 설명서

"버킷 작업"

"PutObject 를 선택합니다"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청과 함께 다음과 같은 추가 헤더가 있습니다.

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

- 개체의 이진 데이터입니다

StorageGRID 설명서

"PutObject 를 선택합니다"

"PutObjectLegalHold를 선택합니다"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

StorageGRID는 구현 시 Amazon S3 REST API에 의해 정의된 모든 요청 본문 매개 변수를 지원합니다.

StorageGRID 설명서

"S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다"

"PutObjectLockConfiguration 을 참조하십시오"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

StorageGRID는 구현 시 Amazon S3 REST API에 의해 정의된 모든 요청 본문 매개 변수를 지원합니다.

StorageGRID 설명서

"S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다"

"PutObjectRetention"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청과 함께 다음 추가 헤더가 추가되었습니다.

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

요청 본문

StorageGRID는 구현 시 Amazon S3 REST API에 의해 정의된 모든 요청 본문 매개 변수를 지원합니다.

StorageGRID 설명서

"S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다"

"PutObjectTagging"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

StorageGRID는 구현 시 Amazon S3 REST API에 의해 정의된 모든 요청 본문 매개 변수를 지원합니다.

StorageGRID 설명서

"객체에 대한 작업"

"RestoreObject 를 선택합니다"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

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

"SelectObjectContent 를 선택합니다"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 요청할 수 있습니다.

요청 본문

지원되는 본문 필드에 대한 자세한 내용은 다음을 참조하십시오.

- "S3 Select를 사용합니다"
- "SelectObjectContent 를 선택합니다"

"업로드 파트"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청과 함께 다음과 같은 추가 URI 쿼리 매개 변수가 추가되었습니다.

- `partNumber`
- `uploadId`

그리고 이러한 추가 요청 헤더는 다음과 같습니다.

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

요청 본문

- 파트의 이진 데이터

StorageGRID 설명서

"업로드 파트"

"업로드파트 복사"

Uri 쿼리 매개 변수 및 요청 헤더

StorageGRID는 모두를 지원합니다 공통 매개 변수 및 머리글 이 요청과 함께 다음과 같은 추가 URI 쿼리 매개 변수가 추가되었습니다.

- `partNumber`
- `uploadId`

그리고 이러한 추가 요청 헤더는 다음과 같습니다.

- `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

없음

StorageGRID 설명서

"업로드파트 복사"

S3 REST API 구성을 테스트합니다

AWS CLI(Amazon Web Services 명령줄 인터페이스)를 사용하여 시스템에 대한 연결을 테스트하고 개체를 읽고 쓸 수 있는지 확인할 수 있습니다.

시작하기 전에

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

- 필요한 경우 가 있습니다 "로드 밸런서 끝점을 생성했습니다". 그렇지 않으면 연결할 스토리지 노드의 IP 주소와 사용할 포트 번호를 알 수 있습니다. 을 참조하십시오 "클라이언트 연결용 IP 주소 및 포트".
- 있습니다 "S3 테넌트 계정을 생성했습니다".
- 테넌트 및 에 로그인했습니다 "선택키를 만들었습니다".

이러한 단계에 대한 자세한 내용은 을 참조하십시오 "클라이언트 연결을 구성합니다".

단계

1. StorageGRID 시스템에서 생성한 계정을 사용하도록 AWS CLI 설정을 구성합니다.
   a. 구성 모드 시작: `aws configure`
   b. 생성한 계정의 액세스 키 ID를 입력합니다.
   c. 생성한 계정의 암호 액세스 키를 입력합니다.
   d. 사용할 기본 영역을 입력합니다. 예를 들면, 다음과 같습니다. `us-east-1`.
   e. 사용할 기본 출력 형식을 입력하거나 * Enter * 를 눌러 JSON을 선택합니다.
2. 버킷을 만듭니다.

   이 예에서는 IP 주소 10.96.101.17 및 포트 10443을 사용하도록 로드 밸런서 끝점을 구성했다고 가정합니다.

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

   버킷이 성공적으로 생성되면 다음 예와 같이 버킷의 위치가 반환됩니다.

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

3. 개체를 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
   ```

   객체가 성공적으로 업로드되면 객체 데이터의 해시인 Etag가 반환됩니다.

4. 버킷의 내용을 나열하여 객체가 업로드되었는지 확인합니다.

   ```
   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
```

StorageGRID에서 S3 REST API를 구현하는 방법

클라이언트 요청 충돌

동일한 키에 쓰는 두 클라이언트 등의 충돌하는 클라이언트 요청은 "최신 성공" 기준으로 해결됩니다.

"Latest-WINS" 평가 시기는 S3 클라이언트가 작업을 시작할 때가 아니라 StorageGRID 시스템이 지정된 요청을 완료하는 시점을 기준으로 합니다.

일관성 값

정합성 보장은 서로 다른 스토리지 노드 및 사이트에서 객체의 가용성과 객체 일관성 간의 균형을 제공합니다. 애플리케이션에 필요한 만큼 일관성을 변경할 수 있습니다.

기본적으로 StorageGRID는 새로 생성된 개체에 대해 쓰기 후 읽기 일관성을 보장합니다. 성공적으로 완료된 PUT를 팔로우하면 새로 작성된 데이터를 읽을 수 있습니다. 기존 오브젝트, 메타데이터 업데이트 및 삭제를 덮어쓰는 것은 결국 일관성이 유지됩니다. 덮어쓰기는 일반적으로 전파되는 데 몇 초 또는 몇 분이 걸리지만 최대 15일이 소요될 수 있습니다.

개체 작업을 다른 일관성으로 수행하려는 경우 다음을 수행할 수 있습니다.

- 에 대한 일관성을 지정합니다 각 버킷.
- 에 대한 일관성을 지정합니다 각 API 작동.
- 다음 작업 중 하나를 수행하여 그리드 전체의 기본 일관성을 변경합니다.
  - 그리드 관리자에서 * 구성 * > * 시스템 * > * 스토리지 설정 * > * 기본 일관성 * 로 이동합니다.
  - .

> 그리드 전체의 일관성에 대한 변경은 설정이 변경된 후에 생성된 버킷에만 적용됩니다. 변경에 대한 세부 정보를 확인하려면 에 있는 감사 로그를 참조하십시오 `/var/local/log` (* consistencyLevel* 검색).

일관성 값

일관성은 StorageGRID이 오브젝트를 추적하는 데 사용하는 메타데이터가 노드 간에 분산되는 방식과 클라이언트

요청을 위한 개체의 가용성에 영향을 줍니다.

버킷 또는 API 작업의 정합성을 다음 값 중 하나로 설정할 수 있습니다.

- * ALL *: 모든 노드가 즉시 데이터를 수신하거나 요청이 실패합니다.
- * 강력한 글로벌 *: 모든 사이트에서 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다.
- * 강력한 사이트 *: 사이트 내의 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다.
- *Read-after-new-write *: (기본값) 새 개체에 대해 읽기-쓰기 후 일관성을 제공하고 개체 업데이트에 대한 최종 일관성을 제공합니다. 고가용성 및 데이터 보호 보장 제공 대부분의 경우에 권장됩니다.
- * 사용 가능 *: 새 객체 및 객체 업데이트 모두에 대한 최종 일관성을 제공합니다. S3 버킷의 경우 필요한 경우에만 사용하십시오(예: 거의 읽지 않는 로그 값이 포함된 버킷의 경우 또는 존재하지 않는 키의 헤드 또는 GET 작업의 경우). S3 FabricPool 버킷은 지원되지 않습니다.

"Read-after-new-write" 및 **"Available"** 정합성 보장을 사용합니다

HEAD 또는 GET 작업에서 "Read-after-new-write" 일관성을 사용하는 경우 StorageGRID는 다음과 같이 여러 단계로 조회를 수행합니다.

- 먼저 낮은 일관성을 사용하여 오브젝트를 찾습니다.
- 이 조회가 실패하면 다음 일관성 값에서 조회를 반복하여 강력한 글로벌 동작과 동일한 일관성을 유지합니다.

HEAD 또는 GET 작업에서 "Read-after-new-write" 일관성을 사용하지만 객체가 존재하지 않는 경우 객체 조회는 항상 강력한 글로벌 동작과 동일한 일관성을 유지합니다. 이 일관성을 유지하기 위해서는 각 사이트에서 개체 메타데이터의 여러 복사본을 사용할 수 있어야 하므로, 같은 사이트에 있는 두 개 이상의 스토리지 노드를 사용할 수 없는 경우 500개의 내부 서버 오류가 발생할 수 있습니다.

Amazon S3와 유사한 일관성 보장이 필요하지 않은 경우 일관성을 "사용 가능"으로 설정하여 헤드 및 가져오기 작업에 대한 이러한 오류를 방지할 수 있습니다. 두부 또는 GET 작업에서 "사용 가능한" 일관성을 사용하는 경우 StorageGRID는 최종 일관성을 제공합니다. 일관성 향상을 위해 실패한 작업을 다시 시도하지 않으므로 개체 메타데이터의 여러 복사본을 사용할 필요가 없습니다.

API 작업에 대한 일관성을 지정합니다

개별 API 작업에 대한 일관성을 설정하려면 작업에 대해 정합성 보장 값을 지원해야 하며 요청 헤더에서 일관성을 지정해야 합니다. 이 예제에서는 GetObject 작업의 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 대해 동일한 일관성을 사용해야 합니다.

버킷의 일관성을 지정합니다

StorageGRID를 사용하여 버킷의 일관성을 설정할 수 있습니다 "버킷 일관성을 유지합니다" 요청하십시오. 아니면

가능합니다 "버킷의 일관성을 변경합니다" Tenant Manager에서

버킷의 일관성을 설정할 때 다음 사항에 유의하십시오.

- 버킷의 일관성을 설정하면 버킷의 오브젝트나 버킷 구성에 수행되는 S3 작업에 사용되는 일관성이 결정됩니다. 버킷 자체의 작동에는 영향을 미치지 않습니다.
- 개별 API 작업의 일관성이 버킷의 일관성을 재정의합니다.
- 일반적으로 버킷은 "Read-after-new-write"라는 기본 일관성을 사용해야 합니다. 요청이 제대로 작동하지 않는 경우 가능한 경우 응용 프로그램 클라이언트 동작을 변경합니다. 또는 각 API 요청의 일관성을 지정하도록 클라이언트를 구성합니다. 버킷 수준의 일관성을 마지막 수단으로 설정합니다.

일관성 및 **ILM** 규칙이 데이터 보호에 영향을 미치는 방식

일관성을 선택하고 ILM 규칙을 따르는 것은 오브젝트가 보호되는 방식에 영향을 미칩니다. 이러한 설정은 상호 작용할 수 있습니다.

예를 들어, 오브젝트가 저장될 때 사용되는 일관성은 오브젝트 메타데이터의 초기 배치에 영향을 주고, ILM 규칙에 대해 선택된 수집 동작은 오브젝트 복사본의 초기 배치에 영향을 미칩니다. StorageGRID에서는 클라이언트 요청을 이행하기 위해 오브젝트의 메타데이터와 해당 데이터에 모두 액세스해야 하므로 일관성 및 수집 동작에 대해 일치하는 보호 수준을 선택하면 초기 데이터 보호 수준을 높이고 시스템 응답을 보다 예측 가능하게 할 수 있습니다.

다음 사항을 참조하십시오 "수집 옵션" ILM 규칙에 사용 가능:

이중 커밋

StorageGRID는 즉시 개체의 중간 복사본을 만들고 클라이언트에 성공을 반환합니다. ILM 규칙에 지정된 복사본은 가능한 경우 만들어집니다.

엄격한

ILM 규칙에 지정된 모든 복제본이 클라이언트에 반환되기 전에 만들어져야 합니다.

균형

StorageGRID는 수집 시 ILM 규칙에 지정된 모든 복제본을 만들려고 합니다. 이 작업이 불가능할 경우 중간 복제본이 만들어지고 성공이 클라이언트에 반환됩니다. ILM 규칙에 지정된 복사본은 가능한 경우 만들어집니다.

일관성과 **ILM** 규칙이 상호 작용하는 방법의 예

다음과 같은 ILM 규칙과 다음과 같은 일관성이 있는 2개 사이트 그리드가 있다고 가정합니다.

- * ILM 규칙 *: 로컬 사이트와 원격 사이트에 각각 하나씩, 두 개의 오브젝트 복사본을 만듭니다. 엄격한 수집 동작을 사용합니다.
- * Consistency *: 강력한 글로벌(오브젝트 메타데이터는 모든 사이트에 즉시 배포됨).

클라이언트가 오브젝트를 그리드에 저장할 때 StorageGRID는 오브젝트 복사본을 둘 다 만들고 메타데이터를 두 사이트에 분산한 다음 클라이언트에 성공을 반환합니다.

수집 성공 메시지가 표시된 시점에 객체가 손실로부터 완벽하게 보호됩니다. 예를 들어, 수집 직후 로컬 사이트가 손실되면 오브젝트 데이터와 오브젝트 메타데이터의 복사본이 원격 사이트에 계속 존재합니다. 개체를 완전히 검색할 수 있습니다.

대신 동일한 ILM 규칙과 강력한 사이트 일관성을 사용한 경우 개체 데이터가 원격 사이트에 복제된 후 개체

메타데이터가 이 사이트에 배포되기 전에 클라이언트에서 성공 메시지를 받을 수 있습니다. 이 경우 오브젝트 메타데이터의 보호 수준이 오브젝트 데이터의 보호 수준과 일치하지 않습니다. 수집 후 곧바로 로컬 사이트가 손실되면 오브젝트 메타데이터가 손실됩니다. 개체를 검색할 수 없습니다.

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

오브젝트 버전 관리

각 오브젝트의 여러 버전을 유지하려면 버킷의 버전 관리 상태를 설정할 수 있습니다. 버킷에 대한 버전 관리를 사용하면 실수로 개체가 삭제되지 않도록 보호하고 이전 버전의 개체를 검색 및 복원할 수 있습니다.

StorageGRID 시스템은 대부분의 기능을 지원하는 버전 관리를 구현하지만 몇 가지 제한 사항이 있습니다. StorageGRID는 각 오브젝트의 버전을 최대 1,000개까지 지원합니다.

오브젝트 버전 관리를 StorageGRID ILM(정보 라이프사이클 관리) 또는 S3 버킷 라이프사이클 구성과 결합할 수 있습니다. 각 버킷에 대해 버전 관리를 명시적으로 설정해야 합니다. 버킷에 대해 버전 관리를 사용하도록 설정하면 버킷에 추가된 각 오브젝트에 버전 ID가 할당되며, StorageGRID 시스템에서 생성됩니다.

MFA(다중 요소 인증) 삭제 사용은 지원되지 않습니다.

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

ILM 및 버전 관리

ILM 정책은 개체의 각 버전에 적용됩니다. ILM 스캔 프로세스는 모든 개체를 지속적으로 스캔하고 현재 ILM 정책에 대해 다시 평가합니다. ILM 정책에 대한 모든 변경 사항은 이전에 수집된 모든 개체에 적용됩니다. 여기에는 버전 관리가 활성화된 경우 이전에 수집된 버전이 포함됩니다. ILM 스캐닝은 이전에 수집된 개체에 새로운 ILM 변경 사항을 적용합니다.

버전 관리 지원 버킷의 S3 객체의 경우 버전 관리 지원을 통해 "현재 시간"을 참조 시간으로 사용하는 ILM 규칙을 생성할 수 있습니다. "이전 객체 버전에만 이 규칙을 적용하시겠습니까?"라는 질문에 대해서는 * 예 * 를 선택하십시오. 인치 "ILM 규칙 만들기 마법사의 1단계")를 클릭합니다. 개체가 업데이트되면 이전 버전은 업데이트되지 않습니다. "비현재 시간" 필터를 사용하면 이전 버전의 객체가 스토리지에 미치는 영향을 줄이는 정책을 만들 수 있습니다.

> ⓘ 다중 파트 업로드 작업을 사용하여 새 버전의 개체를 업로드할 때 개체의 원래 버전에 대한 비현재 시간은 다중 파트 업로드가 완료될 때가 아닌 새 버전에 대해 다중 파트 업로드가 생성된 시점을 반영합니다. 제한된 경우 원래 버전의 비현재 시간이 현재 버전의 시간보다 몇 시간 또는 며칠 빨라질 수 있습니다.

관련 정보

- "S3 버전 오브젝트 삭제 방법"
- "S3 버전 오브젝트 ILM 규칙 및 정책(예 4)".

S3 REST API를 사용하여 **S3** 오브젝트 잠금을 구성합니다

StorageGRID 시스템에서 글로벌 S3 오브젝트 잠금 설정이 활성화된 경우 S3 오브젝트 잠금이 활성화된 버킷을 생성할 수 있습니다. 각 오브젝트 버전에 대해 각 버킷의 기본 보존 또는 보존 설정을 지정할 수 있습니다.

버킷에 대해 **S3** 오브젝트 잠금을 활성화하는 방법

StorageGRID 시스템에 대해 글로벌 S3 오브젝트 잠금 설정이 활성화된 경우 각 버킷을 생성할 때 선택적으로 S3 오브젝트 잠금을 활성화할 수 있습니다.

S3 오브젝트 잠금은 버킷을 생성할 때만 활성화할 수 있는 영구 설정입니다. 버킷을 생성한 후에는 S3 오브젝트 잠금을 추가하거나 비활성화할 수 없습니다.

버킷에 대해 S3 오브젝트 잠금을 설정하려면 다음 방법 중 하나를 사용하십시오.

- 테넌트 관리자를 사용하여 버킷을 생성합니다. 을 참조하십시오 "S3 버킷을 생성합니다".
- 과 함께 CreateBucket 요청을 사용하여 버킷을 만듭니다 `x-amz-bucket-object-lock-enabled` 요청 헤더. 을 참조하십시오 "버킷 작업".

S3 오브젝트 잠금에서는 버킷 버전 관리가 필요하며, 이 버전은 버킷을 생성할 때 자동으로 활성화됩니다. 버킷의 버전 관리는 일시 중단할 수 없습니다. 을 참조하십시오 "오브젝트 버전 관리".

버킷의 기본 보존 설정입니다

버킷에 대해 S3 오브젝트 잠금이 활성화된 경우 버킷에 대한 기본 보존을 선택적으로 설정하고 기본 보존 모드 및 기본 보존 기간을 지정할 수 있습니다.

기본 보존 모드

- 규정 준수 모드:
  - 보존 기한 에 도달할 때까지 개체를 삭제할 수 없습니다.
  - 오브젝트의 보존 기한 을 늘릴 수 있지만 줄일 수는 없습니다.
  - 개체의 보존 기한 은 해당 날짜에 도달할 때까지 제거할 수 없습니다.
- 거버넌스 모드:
  - 를 가진 사용자 `s3:BypassGovernanceRetention` 사용 권한은 를 사용할 수 있습니다 `x-amz-bypass-governance-retention: true` 보존 설정을 무시하도록 헤더를 요청합니다.
  - 이러한 사용자는 보존 기한이 되기 전에 개체 버전을 삭제할 수 있습니다.
  - 이러한 사용자는 개체의 보존 기간(Retain-until-date)을 증가, 감소 또는 제거할 수 있습니다.

기본 보존 기간

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

버킷의 기본 보존 설정 방법

버킷의 기본 보존을 설정하려면 다음 방법 중 하나를 사용합니다.

- 테넌트 관리자에서 버킷 설정을 관리합니다. 을 참조하십시오 "S3 버킷을 생성합니다" 및 "S3 Object Lock 기본 보존을 업데이트합니다".
- 버킷에 대한 PutObjectLockConfiguration 요청을 실행하여 기본 모드와 기본 일 또는 년 수를 지정합니다.

PutObjectLockConfiguration 을 참조하십시오

PutObjectLockConfiguration 요청을 사용하면 S3 오브젝트 잠금이 설정된 버킷의 기본 보존 모드 및 기본 보존 기간을 설정하고 수정할 수 있습니다. 이전에 구성한 기본 보존 설정을 제거할 수도 있습니다.

새 오브젝트 버전이 버킷에 수집되면 기본 보존 모드가 적용됩니다(인 경우) `x-amz-object-lock-mode` 및 `x-amz-object-lock-retain-until-date` 지정되지 않았습니다. 기본 보존 기간은 다음 경우에 보존 기간을 계산하는 데 사용됩니다 `x-amz-object-lock-retain-until-date` 이(가) 지정되지 않았습니다.

오브젝트 버전을 수집한 후 기본 보존 기간을 수정하면 오브젝트 버전의 보존 기한은 그대로 유지되고 새 기본 보존 기간을 사용하여 다시 계산되지 않습니다.

에 가 있어야 합니다 `s3:PutBucketObjectLockConfiguration` 이 작업을 완료하려면 권한 또는 계정 루트 권한이 있어야 합니다.

를 클릭합니다 `Content-MD5` 요청 헤더를 PUT 요청에 지정해야 합니다.

요청 예

이 예에서는 버킷에 대해 S3 Object Lock을 설정하고 기본 보존 모드를 규정 준수 로 설정하고 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 결정 방법

버킷에 대해 S3 오브젝트 잠금이 설정되었는지 확인하고 기본 보존 모드 및 보존 기간을 확인하려면 다음 방법 중 하나를 사용하십시오.

- 테넌트 관리자에서 버킷을 확인합니다. 을 참조하십시오 "S3 버킷을 봅니다".

- GetObjectLockConfiguration 요청을 실행합니다.

GetObjectLockConfiguration 을 참조하십시오

GetObjectLockConfiguration 요청을 사용하면 버킷에 대해 S3 오브젝트 잠금이 설정되어 있는지 확인하고, 사용하도록 설정되어 있는 경우 버킷에 대해 구성된 기본 보존 모드 및 보존 기간이 있는지 확인할 수 있습니다.

새 오브젝트 버전이 버킷에 수집되면 기본 보존 모드가 적용됩니다(인 경우) `x-amz-object-lock-mode` 이(가) 지정되지 않았습니다. 기본 보존 기간은 다음 경우에 보존 기간을 계산하는 데 사용됩니다 `x-amz-object-lock-retain-until-date` 이(가) 지정되지 않았습니다.

에 가 있어야 합니다 `s3:GetBucketObjectLockConfiguration` 이 작업을 완료하려면 권한 또는 계정 루트 권한이 있어야 합니다.

요청 예

```
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 예

```
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>
```

개체의 보존 설정을 지정하는 방법

S3 오브젝트 잠금이 활성화된 버킷에는 S3 오브젝트 잠금 보존 설정이 있는 오브젝트와 없는 오브젝트의 조합이 포함될 수 있습니다.

오브젝트 레벨의 보존 설정은 S3 REST API를 사용하여 지정됩니다. 객체에 대한 보존 설정은 버킷의 기본 보존 설정보다 우선합니다.

각 개체에 대해 다음 설정을 지정할 수 있습니다.

- * 보존 모드 *: 규정 준수 또는 거버넌스 중 하나입니다.
- * Retain-until-date *: StorageGRID에서 개체 버전을 유지해야 하는 기간을 지정하는 날짜입니다.
  - 준수 모드에서 보존 기한이 미래인 경우 오브젝트를 검색할 수 있지만 수정하거나 삭제할 수 없습니다. 보관 기한 을 늘릴 수 있지만 이 날짜는 감소 또는 제거할 수 없습니다.
  - 거버넌스 모드에서 특별 권한이 있는 사용자는 보존 기한 설정을 무시할 수 있습니다. 보존 기간이 경과하기 전에 객체 버전을 삭제할 수 있습니다. 또한 보존 기간을 늘리거나 줄이거나 제거할 수도 있습니다.
- * 법적 증거 자료 보관 *: 개체 버전에 법적 증거 자료 보관 기능을 적용하면 해당 개체가 즉시 잠깁니다. 예를 들어 조사 또는 법적 분쟁과 관련된 객체에 법적 보류를 지정해야 할 수 있습니다. 법적 보류는 만료 날짜가 없지만 명시적으로 제거될 때까지 유지됩니다.

  개체에 대한 법적 보류 설정은 보존 모드 및 보존 기한 과 무관합니다. 개체 버전이 법적 증거 자료 보관 중인 경우 해당 버전을 삭제할 수 없습니다.

오브젝트 버전을 버킷에 추가할 때 S3 오브젝트 잠금 설정을 지정하려면 을 실행합니다 "PutObject 를 선택합니다", "CopyObject 를 선택합니다", 또는 "CreateMultptUpload 를 클릭합니다" 요청하십시오.

다음을 사용할 수 있습니다.

- `x-amz-object-lock-mode`규정 준수 또는 거버넌스(대/소문자 구분)일 수 있습니다.

  > 를 지정할 경우 `x-amz-object-lock-mode`, 또한 을 지정해야 합니다 `x-amz-object-lock-retain-until-date`.

- `x-amz-object-lock-retain-until-date`
  - 보존 기간 값은 형식이어야 합니다 `2020-08-10T21:46:00Z`. 소수 자릿수는 허용되지만 소수점 이하 자릿수는 3자리만 유지됩니다(밀리초 단위). 다른 ISO 8601 형식은 허용되지 않습니다.
  - 보존 종료 날짜는 미래여야 합니다.
- `x-amz-object-lock-legal-hold`

  법적 증거 자료 보관(대소문자 구분)이 켜져 있는 경우, 해당 물체는 법적 증거 자료 보관 하에 배치됩니다. 법적 증거 자료 보관 기능이 꺼져 있는 경우 법적 증거 자료 보관 작업이 없습니다. 다른 값을 사용하면 400개의 잘못된 요청(InvalidArgument) 오류가 발생합니다.

이러한 요청 헤더를 사용하는 경우 다음과 같은 제한 사항에 유의하십시오.

- 를 클릭합니다 `Content-MD5` 요청 헤더가 필요한 경우 `x-amz-object-lock-*` PutObject 요청에 요청 헤더가 있습니다. `Content-MD5` CopyObject 또는 CreateMultipartUpload에는 필요하지 않습니다.
- 버킷에 S3 오브젝트 잠금이 설정되어 있지 않은 경우 및 가 활성화되어 있어야 합니다 `x-amz-object-lock-*` 요청 헤더가 있으면 400개의 잘못된 요청(InvalidRequest) 오류가 반환됩니다.
- PutObject 요청에서는 의 사용을 지원합니다 `x-amz-storage-class: REDUCED_REDUNDANCY` AWS 동작과 일치시킵니다. 하지만 오브젝트가 S3 오브젝트 잠금이 설정된 버킷으로 수집되면 StorageGRID는 항상 이중 커밋 수집을 수행합니다.
- 이후의 Get 또는 HeadObject 버전 응답에는 헤더가 포함됩니다 `x-amz-object-lock-mode`, `x-amz-object-lock-retain-until-date`, 및 `x-amz-object-lock-legal-hold`, 구성된 경우 및 요청 보낸 사람이 올바른 경우 `s3:Get*` 권한.

를 사용할 수 있습니다 `s3:object-lock-remaining-retention-days` 개체에 대해 허용되는 최소 및 최대 보존 기간을 제한하는 정책 조건 키입니다.

개체의 보존 설정을 업데이트하는 방법

기존 개체 버전에 대한 법적 증거 자료 보관 또는 보존 설정을 업데이트해야 하는 경우 다음 개체 하위 리소스 작업을 수행할 수 있습니다.

- `PutObjectLegalHold`

  새 법적 증거 자료 보관 값이 켜져 있으면 해당 개체는 법적 증거 자료 보관 아래에 배치됩니다. 법적 증거 자료 보관 가치가 없는 경우 법적 구속이 해제됩니다.

- `PutObjectRetention`
  - 모드 값은 규정 준수 또는 거버넌스(대/소문자 구분)일 수 있습니다.
  - 보존 기간 값은 형식이어야 합니다 `2020-08-10T21:46:00Z`. 소수 자릿수는 허용되지만 소수점 이하 자릿수는 3자리만 유지됩니다(밀리초 단위). 다른 ISO 8601 형식은 허용되지 않습니다.

◦ 개체 버전에 기존 보존 기한이 있는 경우 개체 버전을 늘릴 수만 있습니다. 새 값은 미래여야 합니다.

거버넌스 모드 사용 방법

를 가진 사용자 `s3:BypassGovernanceRetention` 권한은 거버넌스 모드를 사용하는 개체의 활성 보존 설정을 무시할 수 있습니다. 삭제 또는 PutObjectRetention 작업에는 가 포함되어야 합니다 `x-amz-bypass-governance-retention:true` 요청 헤더. 이러한 사용자는 다음과 같은 추가 작업을 수행할 수 있습니다.

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

  법적 증거 자료 보관 중인 객체는 삭제할 수 없습니다. 법적 증거 자료 보관 기능을 해제해야 합니다.

- 개체의 보존 기간이 경과하기 전에 개체 버전의 모드를 거버넌스에서 규정 준수로 변경하는 PutObjectRetention 작업을 수행합니다.

  규정 준수 모드를 거버넌스로 변경하는 것은 허용되지 않습니다.

- PutObjectRetention 작업을 수행하여 개체 버전의 보존 기간을 증가, 감소 또는 제거합니다.

관련 정보

- "S3 오브젝트 잠금으로 오브젝트 관리"
- "S3 오브젝트 잠금을 사용하여 오브젝트를 보존합니다"
- "Amazon Simple Storage Service 사용자 가이드: S3 Object Lock 사용"

S3 라이프사이클 구성을 생성합니다

S3 라이프사이클 구성을 생성하여 StorageGRID 시스템에서 특정 오브젝트 삭제 시기를 제어할 수 있습니다.

이 섹션의 간단한 예는 S3 라이프사이클 구성에서 특정 S3 버킷에서 특정 객체가 삭제(만료)되는 시기를 제어하는 방법을 보여줍니다. 이 섹션의 예제는 설명을 위한 것입니다. S3 라이프사이클 구성 생성에 대한 자세한 내용은 를 참조하십시오 "Amazon Simple Storage Service 사용자 가이드: 객체 수명 주기 관리". StorageGRID는 만료 작업만 지원하며 전환 작업은 지원하지 않습니다.

문서 수정 상태 설정은 무엇입니까

라이프사이클 구성은 특정 S3 버킷의 오브젝트에 적용되는 규칙 세트입니다. 각 규칙은 영향을 받는 개체와 해당 개체가 만료되는 시기(특정 날짜 또는 특정 일 수 이후)를 지정합니다.

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

- 만료: 지정된 날짜에 도달하거나 지정된 일 수에 도달할 때 개체를 인제스트할 때로부터 개체를 삭제합니다.
- NoncurrentVersionExpiration: 지정된 일 수에 도달할 때 개체가 비전류가 되었을 때부터 개체를 삭제합니다.
- 필터(접두사, 태그)
- 상태
- ID입니다

각 오브젝트는 S3 버킷 라이프사이클 또는 ILM 정책의 보존 설정을 따릅니다. S3 버킷 라이프사이클이 구성되면 라이프사이클 만료 작업이 버킷 라이프사이클 필터와 일치하는 오브젝트에 대한 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 설정이 사용되지 않으며 객체 버전이 영구적으로 보존됩니다. 을 참조하십시오 "S3 버킷 라이프사이클 및 ILM 정책의 우선순위 예".

따라서 ILM 규칙의 배치 지침이 개체에 계속 적용되더라도 그리드에서 개체를 제거할 수 있습니다. 또는 개체에 대한 ILM 배치 지침이 만료된 후에도 개체가 그리드에 남아 있을 수 있습니다. 자세한 내용은 을 참조하십시오 "ILM이 개체 수명 전반에 걸쳐 작동하는 방식".

> 버킷 수명 주기 구성은 S3 오브젝트 잠금이 활성화된 버킷과 함께 사용할 수 있지만 버킷 수명 주기 구성은 레거시 준수 버킷에서 지원되지 않습니다.

StorageGRID는 다음 버킷 작업을 사용하여 라이프사이클 구성을 관리합니다.

- DeleteBucketLifecycle
- GetBuckLifecycleConfiguration 을 참조하십시오
- PutBucketLifecycleConfiguration을 참조하십시오

문서 수정 상태 설정 작성

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

1. 규칙 1은 접두사와 일치하는 객체에만 적용됩니다 `category1/` 및 이(가) 있습니다 `key2` 의 값 `tag2`. 를 클릭합니다 `Expiration` 매개 변수는 필터와 일치하는 개체가 2020년 8월 22일 자정에 만료되도록 지정합니다.
2. 규칙 2는 접두사와 일치하는 객체에만 적용됩니다 `category2/`. 를 클릭합니다 `Expiration` 매개 변수는 필터와 일치하는 개체가 수집된 후 100일이 경과하도록 지정합니다.

   > 일 수를 지정하는 규칙은 오브젝트가 수집된 시점을 기준으로 합니다. 현재 날짜가 수집 날짜와 일 수를 더한 값을 초과하면 라이프사이클 구성이 적용되는 즉시 일부 객체가 버킷에서 제거될 수 있습니다.

3. 규칙 3은 접두사와 일치하는 개체에만 적용됩니다 `category3/`. 를 클릭합니다 `Expiration` 매개 변수 일치하는 개체의 현재 버전이 아닌 버전이 최신 상태가 아닌 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 라이프사이클 구성을 적용합니다

문서 수정 상태 구성 파일을 만든 후 PutBucketLifecycleConfiguration 요청을 실행하여 버킷에 적용합니다.

이 요청은 예제 파일의 문서 수정 상태 구성을 이름이 인 버킷의 오브젝트에 적용합니다 `testbucket`.

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

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

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

성공적으로 응답하면 방금 적용한 문서 수정 상태 설정이 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 적용될 경우 응답에는 가 포함됩니다 `Expiration` 객체가 만료되는 시간과 일치하는 만료 규칙을 나타내는 매개 변수입니다.

> 버킷 라이프사이클이 ILM, 을 무시하기 때문입니다 `expiry-date` 객체가 삭제될 실제 날짜가 표시됩니다. 자세한 내용은 을 참조하십시오 "개체 보존이 결정되는 방식".

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

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

성공 응답은 개체가 100일(2020년 10월 1일) 내에 만료되고 라이프사이클 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 testbucket의 동일한 객체에 대한 메타데이터를 가져오는 데 사용되었습니다.

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

성공 응답에는 개체의 메타데이터가 포함되며 개체가 100일 후에 만료되고 규칙 2와 일치함을 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 관리 지원 버킷의 경우 `x-amz-expiration` 응답 헤더는 개체의 현재 버전에만 적용됩니다.

S3 REST API 구현을 위한 권장사항

StorageGRID와 함께 사용할 S3 REST API를 구현할 때는 다음 권장 사항을 따라야 합니다.

존재하지 않는 객체에 대한 헤드 권장 사항

응용 프로그램에서 개체가 실제로 존재한다고 예상하지 않는 경로에 있는지 정기적으로 확인하는 경우 "사용 가능"을 사용해야 합니다. "정합성". 예를 들어, 응용 프로그램이 해당 위치에 배치하기 전에 해당 위치를 헤딩하는 경우 "사용 가능한" 일관성을 사용해야 합니다.

그렇지 않으면 헤드 작업에서 객체를 찾지 못할 경우 같은 사이트에 있는 두 개 이상의 스토리지 노드를 사용할 수 없거나 원격 사이트에 연결할 수 없는 경우 500개의 내부 서버 오류가 발생할 수 있습니다.

을 사용하여 각 버킷에 대해 "사용 가능한" 일관성을 설정할 수 있습니다 "버킷 일관성을 유지합니다" 요청 또는 개별 API 작업에 대한 요청 헤더에서 일관성을 지정할 수 있습니다.

개체 키에 대한 권장 사항

버킷이 처음 생성된 시점을 기준으로 오브젝트 키 이름에 대한 다음 권장 사항을 따르십시오.

StorageGRID 11.4 또는 이전 버전에서 생성된 버킷

- 개체 키의 처음 네 문자로 임의 값을 사용하지 마십시오. 이는 이전 AWS에서 권장하는 키 접두사와 다릅니다. 대신 와 같이 고유하지 않은 접두어를 사용합니다 `image`.
- 이전 AWS 권장 사항에 따라 키 접두사에 랜덤 및 고유 문자를 사용하려면 오브젝트 키에 디렉토리 이름이 접두사로 지정됩니다. 즉, 다음 형식을 사용합니다.

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

이 형식 대신:

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

StorageGRID 11.4 이상에서 생성된 버킷

성능 모범 사례에 맞게 개체 키 이름을 제한하는 것은 필요하지 않습니다. 대부분의 경우 개체 키 이름의 처음 4개 문자에 임의의 값을 사용할 수 있습니다.

> 단, 짧은 시간 내에 모든 오브젝트를 지속적으로 제거하는 S3 워크로드가 예외입니다. 이 사용 사례에 대한 성능 영향을 최소화하려면 키와 같은 1000개의 오브젝트마다 주요 이름의 앞부분을 다르게 지정해야 합니다. 예를 들어, S3 클라이언트가 일반적으로 초당 2,000개의 오브젝트를 기록하고 ILM 또는 버킷 라이프사이클 정책에 따라 3일 후에 모든 오브젝트를 제거한다고 가정해 보겠습니다. 성능에 미치는 영향을 최소화하기 위해 다음과 같은 패턴을 사용하여 키의 이름을 지정할 수 있습니다.
>
> `/mybucket/mydir/yyyymmddhhmmss-random_UUID.jpg`

"범위 읽기**"**에 대한 권장 사항

를 누릅니다 "저장된 개체를 압축하는 전역 옵션" 이 활성화되면 S3 클라이언트 응용 프로그램은 반환되는 바이트 범위를 지정하는 GetObject 작업을 수행하지 않아야 합니다. 이러한 "범위 읽기" 작업은 StorageGRID에서 요청된 바이트에 액세스하기 위해 개체의 압축을 효과적으로 해제해야 하기 때문에 비효율적입니다. 매우 큰 개체에서 작은 범위의 바이트를 요청하는 GetObject 작업은 특히 비효율적입니다. 예를 들어, 50GB의 압축된 개체에서 10MB 범위를 읽는 것은 비효율적입니다.

압축된 개체에서 범위를 읽으면 클라이언트 요청이 시간 초과될 수 있습니다.

> 개체를 압축해야 하고 클라이언트 응용 프로그램에서 범위 읽기를 사용해야 하는 경우 응용 프로그램의 읽기 시간 초과를 늘리십시오.

Amazon S3 REST API 지원

S3 REST API 구현 세부 정보

StorageGRID 시스템은 대부분의 작업을 지원하고 몇 가지 제한 사항이 있는 간단한 스토리지 서비스 API(API 버전 2006-03-01)를 구현합니다. S3 REST API 클라이언트 애플리케이션을 통합할 때 구현 세부 정보를 이해해야 합니다.

StorageGRID 시스템은 가상 호스팅 방식의 요청과 경로 스타일 요청을 모두 지원합니다.

날짜 처리

S3 REST API의 StorageGRID 구현은 유효한 HTTP 날짜 형식만 지원합니다.

StorageGRID 시스템은 날짜 값을 허용하는 모든 헤더에 대해 유효한 HTTP 날짜 형식만 지원합니다. 날짜의 시간 부분은 그리니치 표준시(GMT) 형식 또는 표준 시간대 오프셋 없이 UTC(국제 표준시) 형식으로 지정할 수 있습니다(+0000을 지정해야 함). 을 포함하는 경우 `x-amz-date` 헤더 요청의 날짜 요청 헤더에 지정된 모든 값을 재정의합니다. AWS 서명 버전 4를 사용하는 경우 `x-amz-date` 날짜 헤더가 지원되지 않으므로 서명된 요청에 헤더가 있어야 합니다.

공통 요청 헤더

StorageGRID 시스템은 에 정의된 공통 요청 헤더를 지원합니다 "Amazon Simple Storage Service API 참조: 일반 요청 헤더"한 가지 예외가 있습니다.

요청 헤더	구축
권한 부여	AWS Signature 버전 2에 대한 전체 지원 다음 경우를 제외하고 AWS Signature 버전 4 지원: • 요청 본문에 대한 SHA256 값이 계산되지 않습니다. 사용자가 제출한 값은 마치 값이 있는 것처럼 유효성 검사 없이 승인됩니다 `UNSIGNED-PAYLOAD` 에 대한 정보가 제공되었습니다 `x-amz-content-sha256` 머리글.
X-amz-security-token	구현되지 않았습니다. 반환 `XNotImplemented`.

공통 응답 헤더

StorageGRID 시스템은 한 가지 예외를 제외하고 _Simple Storage Service API Reference_에 의해 정의된 모든 공통 응답 헤더를 지원합니다.

응답 헤더	구축
X-amz-id-2	사용 안 합니다

요청을 인증합니다

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

S3 API는 S3 API 요청을 인증하는 데 서명 버전 2 및 서명 버전 4를 지원합니다.

인증된 요청은 액세스 키 ID 및 비밀 액세스 키를 사용하여 서명해야 합니다.

StorageGRID 시스템은 HTTP라는 두 가지 인증 방법을 지원합니다 `Authorization` 머리글 및 쿼리 매개 변수 사용

HTTP 인증 헤더를 사용합니다

HTTP `Authorization` 헤더는 버킷 정책에서 허용하는 익명 요청을 제외한 모든 S3 API 작업에서 사용됩니다. 를 클릭합니다 `Authorization` Header 요청을 인증하는 데 필요한 모든 서명 정보를 포함합니다.

쿼리 매개 변수를 사용합니다

쿼리 매개 변수를 사용하여 URL에 인증 정보를 추가할 수 있습니다. 이를 URL 사전 서명 이라고 하며, 이 URL을 사용하여 특정 리소스에 대한 임시 액세스 권한을 부여할 수 있습니다. 미리 지정된 URL을 가진 사용자는 리소스에 액세스하기 위해 비밀 액세스 키를 알 필요가 없으며, 이를 통해 타사에 리소스에 대한 제한된 액세스를 제공할 수 있습니다.

서비스에 대한 작업

StorageGRID 시스템은 서비스에 대해 다음 작업을 지원합니다.

작동	구축
ListBucket (이전 명칭: Get Service)	모든 Amazon S3 REST API 동작으로 구현됩니다. 예고 없이 변경될 수 있습니다.
스토리지 사용량을 가져옵니다	StorageGRID입니다 "스토리지 사용량을 가져옵니다" 요청 은 계정에 의해 사용 중인 총 저장소 양 및 계정과 연결된 각 버킷에 대해 알려줍니다. 이 작업은 /path 및 사용자 지정 쿼리 매개 변수가 있는 서비스에 대한 작업입니다 (`?x-ntap-sg-usage`)가 추가되었습니다.
옵션 /	클라이언트 응용 프로그램을 실행할 수 있습니다 `OPTIONS /` 스토리지 노드의 사용 가능 여부를 결정하기 위해 S3 인증 자격 증명을 제공하지 않고 스토리지 노드의 S3 포트에 대한 요청입니다. 이 요청을 사용하여 모니터링을 수행하거나, 외부 로드 밸런서가 스토리지 노드가 다운된 시점을 식별하도록 할 수 있습니다.

버킷 작업

StorageGRID 시스템은 각 S3 테넌트 계정에 대해 최대 1,000개의 버킷을 지원합니다.

버킷 이름 제한은 AWS US 표준 지역 제한을 따르지만, S3 가상 호스팅 스타일 요청을 지원하기 위해 DNS 명명 규칙으로 제한해야 합니다.

자세한 내용은 다음을 참조하십시오.

- "Amazon Simple Storage Service 사용자 가이드: 버킷 제한 및 제한 사항"
- "S3 끝점 도메인 이름을 구성합니다"

ListObjects(Get Bucket) 및 ListObjectVersions(Get Bucket 개체 버전) 작업은 StorageGRID를 지원합니다 "일관성 값".

개별 버킷에 대해 마지막 액세스 시간에 대한 업데이트가 설정되었는지 여부를 확인할 수 있습니다. 을 참조하십시오 "버킷 최종 액세스 시간 가져오기".

다음 표에서는 StorageGRID에서 S3 REST API 버킷 작업을 구축하는 방법을 설명합니다. 이러한 작업을 수행하려면 계정에 필요한 액세스 자격 증명을 제공해야 합니다.

작동	구축
CreateBucket	새 버킷을 생성합니다. 버킷을 만들면 버켓 소유자가 됩니다. • 버킷 이름은 다음 규칙을 준수해야 합니다. ◦ 각 StorageGRID 시스템에서 고유해야 합니다(테넌트 계정에서만 고유한 것은 아님). ◦ DNS를 준수해야 합니다. ◦ 3자 이상 63자 이하여야 합니다. ◦ 인접한 레이블이 마침표로 구분된 하나 이상의 레이블일 수 있습니다. 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 요청에서 기간을 사용하지 않아야 합니다. 마침표는 서버 와일드카드 인증서 확인에 문제를 일으킬 수 있습니다. • 기본적으로 버킷은 에서 생성됩니다 `us-east-1` 지역. 그러나 을 사용할 수 있습니다 `LocationConstraint` 다른 영역을 지정할 요청 본문의 요청 요소입니다. 를 사용할 때 `LocationConstraint` 요소, 그리드 관리자 또는 그리드 관리 API를 사용하여 정의된 영역의 정확한 이름을 지정해야 합니다. 사용할 지역 이름을 모르는 경우 시스템 관리자에게 문의하십시오. 참고: CreateBucket 요청이 StorageGRID에 정의되지 않은 영역을 사용하는 경우 오류가 발생합니다. • 을 포함할 수 있습니다 `x-amz-bucket-object-lock-enabled` S3 오브젝트 잠금이 활성화된 버킷을 생성하도록 헤더를 요청합니다. 을 참조하십시오 "S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다". 버킷을 생성할 때 S3 오브젝트 잠금을 활성화해야 합니다. 버킷을 생성한 후에는 S3 오브젝트 잠금을 추가하거나 비활성화할 수 없습니다. S3 오브젝트 잠금에는 버킷 버전 관리가 필요하며, 이 버전은 버킷을 생성할 때 자동으로 활성화됩니다.
삭제 버킷	버킷을 삭제합니다.
DeleteBucketCors	버킷에 대한 CORS 구성을 삭제합니다.
DeleteBucketEncryption	버킷에서 기본 암호화를 삭제합니다. 암호화된 기존 개체는 암호화된 상태로 유지되지만 버킷에 추가된 새 개체는 암호화되지 않습니다.
DeleteBucketLifecycle	버킷에서 문서 수정 상태 설정을 삭제합니다. 을 참조하십시오 "S3 라이프사이클 구성을 생성합니다".
DeleteBuckketPolicy를 참조하십시오	버킷에 연결된 정책을 삭제합니다.

작동	구축
DeleteBuckReplication 을 참조하십시오	버킷에 연결된 복제 구성을 삭제합니다.
삭제 BucketTagging	를 사용합니다 `tagging` 버킷에서 모든 태그를 제거하는 하위 리소스입니다. 주의: 이 버킷에 대해 기본값이 아닌 ILM 정책 태그가 설정된 경우 이 있습니다 `NTAP-SG-ILM-BUCKET-TAG` 값이 할당된 버킷 태그입니다. 이 있는 경우 DeleteBucketTagging 요청을 실행하지 마십시오 `NTAP-SG-ILM-BUCKET-TAG` 버킷 태그 대신 만 사용하여 PutBucketTagging 요청을 실행합니다 `NTAP-SG-ILM-BUCKET-TAG` 태그와 지정된 값을 사용하여 버킷에서 다른 모든 태그를 제거합니다. 를 수정하거나 제거하지 마십시오 `NTAP-SG-ILM-BUCKET-TAG` 버킷 태그
GetBuckketAcl	양수 응답과 ID, DisplayName 및 버킷 소유자의 사용 권한을 반환합니다. 이는 소유자가 버킷에 대한 모든 액세스 권한을 가지고 있음을 나타냅니다.
GetBuckketCors 를 참조하십시오	를 반환합니다 `cors` 버킷에 대한 구성.
GetBuckketEncryption을 참조하십시오	버킷의 기본 암호화 구성을 반환합니다.
GetBuckLifecycleConfiguration 을 참조하십시오 (이전 명칭 Get Bucket 수명주기)	버킷의 수명주기 구성을 반환합니다. 을 참조하십시오 "S3 라이프사이클 구성을 생성합니다".
GetBucketLocation 을 참조하십시오	를 사용하여 설정한 영역을 반환합니다 `LocationConstraint` CreateBucket 요청의 요소입니다. 버킷 영역이 인 경우 `us-east-1`영역에 대해 빈 문자열이 반환됩니다.
GetBuckNotificationConfiguration 을 참조하십시오 (이전 명칭 Get Bucket 알림)	버킷에 연결된 알림 구성을 반환합니다.
GetBuckketPolicy를 참조하십시오	버킷에 연결된 정책을 반환합니다.
GetBucketReplication 을 참조하십시오	버킷에 연결된 복제 구성을 반환합니다.

작동	구축
GetBucketTagging	를 사용합니다 `tagging` 버킷에 대한 모든 태그를 반환하는 하위 리소스입니다. 주의: 이 버킷에 대해 기본값이 아닌 ILM 정책 태그가 설정된 경우 이 있습니다 `NTAP-SG-ILM-BUCKET-TAG` 값이 할당된 버킷 태그입니다. 이 태그를 수정하거나 제거하지 마십시오.
GetBuckketVersioning 을 참조하십시오	이 구현에서는 을 사용합니다 `versioning` 버킷의 버전 관리 상태를 반환하는 하위 리소스입니다. • *blank*: 버전 관리가 활성화되지 않았습니다(버킷이 "버전 없음"). • 사용: 버전 관리가 활성화됩니다 • 일시 중단됨: 버전 관리가 이전에 활성화되었으며 일시 중단되었습니다
GetObjectLockConfiguration 을 참조하십시오	구성된 경우 버킷 기본 보존 모드와 기본 보존 기간을 반환합니다. 을 참조하십시오 "S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다".
머리버킷	버킷이 존재하는지, 버킷에 액세스할 수 있는 권한이 있는지 확인합니다. 이 작업은 다음을 반환합니다. • `x-ntap-sg-bucket-id`UUID 형식의 버킷 UUID입니다. • `x-ntap-sg-trace-id`: 연결된 요청의 고유한 추적 ID입니다.
ListObjects 및 ListObjectsV2 를 참조하십시오 (이전 명칭 Get Bucket)	버킷에 있는 오브젝트의 일부 또는 전체(최대 1,000개)를 반환합니다. 오브젝트를 에 인제스트한 경우에도 오브젝트에 대한 스토리지 클래스는 두 값 중 하나를 가질 수 있습니다 `REDUCED_REDUNDANCY` 스토리지 클래스 옵션: • `STANDARD`는 객체가 스토리지 노드로 구성된 스토리지 풀에 저장되었음을 나타냅니다. • `GLACIER`가 표시됩니다. 이는 해당 객체가 Cloud Storage Pool에 지정된 외부 버킷으로 이동되었음을 나타냅니다. 버킷에 동일한 접두사가 있는 삭제된 키의 많은 수가 포함된 경우 응답에 몇 가지 항목이 포함될 수 있습니다 `CommonPrefixes` 여기에는 키가 포함되어 있지 않습니다.
ListObjectVersions 를 선택합니다 (이전에 명명된 Get Bucket Object 버전)	버킷에 대한 읽기 액세스 권한이 있는 경우 이 작업을 와 함께 사용합니다 `versions` 하위 리소스는 버킷에 있는 모든 버전의 오브젝트의 메타데이터를 나열합니다.

작동	구축
BuckketCors의	버킷이 오리진 간 요청을 처리할 수 있도록 버킷에 대한 CORS 구성을 설정합니다. CORS(Cross-origin Resource Sharing)는 한 도메인의 클라이언트 웹 애플리케이션이 다른 도메인의 리소스에 액세스할 수 있도록 하는 보안 메커니즘입니다. 예를 들어, 이라는 S3 버킷을 사용한다고 가정합니다 `images` 그래픽을 저장합니다. 에 대한 CORS 구성을 설정합니다 `images` 버킷을 사용하면 버켓의 이미지를 웹 사이트에 표시할 수 있습니다 `http://www.example.com`.
PutBucketEncryption을 참조하십시오	기존 버킷의 기본 암호화 상태를 설정합니다. 버킷 수준 암호화가 활성화된 경우 버킷에 추가된 모든 새 오브젝트는 암호화됩니다. StorageGRID는 StorageGRID 관리 키로 서버 측 암호화를 지원합니다. 서버 측 암호화 구성 규칙을 지정할 때 를 설정합니다 `SSEAlgorithm` 매개 변수 대상 `AES256`를 사용하지 마십시오 `KMSMasterKeyID` 매개 변수. 객체 업로드 요청이 이미 암호화를 지정한 경우(즉, 요청에 가 포함된 경우) 버킷 기본 암호화 구성은 무시됩니다 `x-amz-server-side-encryption-*` 요청 헤더 참조).
PutBucketLifecycleConfiguration을 참조하십시오 (이전에 명명된 Put Bucket 수명 주기)	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) • 비currentVersionExpiration(NewerNoncurrentVersions, NoncurrentDays) • 필터(접두사, 태그) • 상태 • ID입니다 StorageGRID는 다음 작업을 지원하지 않습니다. • AbortIncompleteMultipartUpload를 중단합니다 • 전환 을 참조하십시오 "S3 라이프사이클 구성을 생성합니다". 버킷 수명 주기의 만료 작업이 ILM 배치 지침과 상호 작용하는 방법을 이해하려면 을 참조하십시오 "ILM이 개체 수명 전반에 걸쳐 작동하는 방식". • 참고 *: 버킷 수명 주기 구성은 S3 오브젝트 잠금이 활성화된 버킷과 함께 사용할 수 있지만 레거시 준수 버킷에서는 버킷 수명 주기 구성이 지원되지 않습니다.

작동	구축

PutBucketNotificationConfiguration을 참조하십시오
 	
(이전에 명명된 Put Bucket 알림)
	요청 본문에 포함된 알림 구성 XML을 사용하여 버킷에 대한 알림을 구성합니다. 다음과 같은 구현 세부 사항에 유의해야 합니다.
	
	• StorageGRID는 Amazon SNS(Simple Notification Service) 또는 Kafka 토픽을 대상으로 지원합니다. SQS(Simple Queue Service) 또는 Amazon Lambda 엔드포인트는 지원되지 않습니다.
	• 알림 대상은 StorageGRID 엔드포인트의 URN으로 지정해야 합니다. 테넌트 관리자 또는 테넌트 관리 API를 사용하여 엔드포인트를 생성할 수 있습니다.
	
	알림 설정을 성공적으로 하려면 끝점이 있어야 합니다. 끝점이 없는 경우, 를 클릭합니다 `400 Bad Request` 코드와 함께 오류가 반환됩니다 `InvalidArgument`.
	
	• 다음 이벤트 유형에 대한 알림을 구성할 수 없습니다. 이러한 이벤트 유형은 * 지원되지 않습니다 *.
	◦ `s3:ReducedRedundancyLostObject`
	◦ `s3:ObjectRestore:Completed`
	• StorageGRID에서 보낸 이벤트 알림은 다음 목록에 표시된 것처럼 일부 키를 포함하지 않고 다른 키에 대해 특정 값을 사용한다는 점을 제외하고 표준 JSON 형식을 사용합니다.
	◦ * eventSource * 를 선택합니다
	
	`sgws:s3`
	
	◦ * awsRegion *
	
	포함되지 않음
	
	◦ x-amz-id-2 *
	
	포함되지 않음
	
	◦ * 표시 *
	
	`urn:sgws:s3:::bucket_name`

BucketPolicy를 참조하십시오	버킷에 연결된 정책을 설정합니다. 을 참조하십시오 "버킷 및 그룹 액세스 정책을 사용합니다".

작동	구축
PutBucketReplication을 참조하십시오	를 구성합니다 "StorageGRID CloudMirror 복제" 요청 본문에 제공된 복제 구성 XML을 사용하는 버킷의 경우 CloudMirror 복제의 경우 다음과 같은 구축 세부 정보를 알고 있어야 합니다. • StorageGRID는 복제 구성의 V1만 지원합니다. 즉, StorageGRID는 의 사용을 지원하지 않습니다 `Filter` 규칙에 대한 요소로, 개체 버전 삭제에 대한 V1 규칙을 따릅니다. 자세한 내용은 을 참조하십시오 "Amazon Simple Storage Service 사용 설명서: 복제 구성". • 버킷 복제는 버전 관리되거나 버전이 지정되지 않은 버킷에서 구성할 수 있습니다. • 복제 구성 XML의 각 규칙에서 다른 대상 버킷을 지정할 수 있습니다. 소스 버킷은 둘 이상의 대상 버킷에 복제할 수 있습니다. • 대상 버킷은 테넌트 관리자 또는 테넌트 관리 API에 지정된 StorageGRID 엔드포인트의 URN으로 지정해야 합니다. 을 참조하십시오 "CloudMirror 복제를 구성합니다". 복제 구성이 성공하려면 엔드포인트가 있어야 합니다. 엔드포인트가 없으면 요청이 로 실패합니다 `400 Bad Request`. 오류 메시지는 다음과 같습니다. `Unable to save the replication policy. The specified endpoint URN does not exist: URN`. • 을 지정할 필요가 없습니다 `Role` 구성 XML에서. 이 값은 StorageGRID에서 사용되지 않으며 제출될 경우 무시됩니다. • 구성 XML에서 스토리지 클래스를 생략하면 StorageGRID에서 를 사용합니다 `STANDARD` 기본적으로 스토리지 클래스입니다. • 소스 버킷에서 객체를 삭제하거나 소스 버킷 자체를 삭제하는 경우 지역 간 복제 동작은 다음과 같습니다. ◦ 복제되기 전에 오브젝트 또는 버킷을 삭제하면 객체/버킷이 복제되지 않으므로 사용자에게 통지되지 않습니다. ◦ 복제된 후 오브젝트 또는 버킷을 삭제하면 StorageGRID는 지역 간 복제 V1에 대한 표준 Amazon S3 삭제 동작을 따릅니다.

작동	구축
BucketTagging	를 사용합니다 `tagging` 하위 리소스로서 버킷에 대한 태그 집합을 추가하거나 업데이트합니다. 버킷 태그를 추가할 때 다음과 같은 제한 사항을 숙지하십시오. • StorageGRID 및 Amazon S3 모두 각 버킷당 최대 50개의 태그를 지원합니다. • 버킷과 연결된 태그에는 고유한 태그 키가 있어야 합니다. 태그 키의 길이는 최대 128자의 유니코드 문자일 수 있습니다. • 태그 값의 길이는 최대 256자의 유니코드 문자일 수 있습니다. • 키와 값은 대/소문자를 구분합니다. 주의: 이 버킷에 대해 기본값이 아닌 ILM 정책 태그가 설정된 경우 이 있습니다 `NTAP-SG-ILM-BUCKET-TAG` 값이 할당된 버킷 태그입니다. 를 확인합니다 `NTAP-SG-ILM-BUCKET-TAG` 버킷 태그는 모든 PutBucketTagging 요청에 할당된 값과 함께 포함됩니다. 이 태그를 수정하거나 제거하지 마십시오. 참고: 이 작업은 버킷에 이미 있는 현재 태그를 덮어씁니다. 기존 태그를 세트에서 생략하면 해당 태그가 버킷에 대해 제거됩니다.
PutBucketVersioning을 참조하십시오	를 사용합니다 `versioning` 기존 버킷의 버전 관리 상태를 설정하는 하위 리소스입니다. 다음 값 중 하나를 사용하여 버전 관리 상태를 설정할 수 있습니다. • Enabled(사용): 버킷의 오브젝트에 대한 버전 관리를 활성화합니다. 버킷에 추가된 모든 오브젝트는 고유한 버전 ID를 받습니다. • Suspended(일시 중지됨): 버킷의 오브젝트에 대한 버전 관리를 비활성화합니다. 버킷에 추가된 모든 오브젝트는 버전 ID를 수신합니다 `null`.
PutObjectLockConfiguration 을 참조하십시오	버킷 기본 보존 모드 및 기본 보존 기간을 구성하거나 제거합니다. 기본 보존 기간이 수정되면 기존 개체 버전의 보존 기한은 그대로 유지되며 새 기본 보존 기간을 사용하여 다시 계산되지 않습니다. 을 참조하십시오 "S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다" 을 참조하십시오.

객체에 대한 작업

객체에 대한 작업

이 섹션에서는 StorageGRID 시스템이 객체에 대해 S3 REST API 작업을 구축하는 방법에 대해 설명합니다.

다음 조건은 모든 개체 작업에 적용됩니다.

- StorageGRID "일관성 값" 는 다음과 같은 경우를 제외하고 모든 개체 작업에서 지원됩니다.
  - GetObjectAcl
  - `OPTIONS /`

- PutObjectLegalHold를 선택합니다
- PutObjectRetention
- SelectObjectContent 를 선택합니다

- 동일한 키에 쓰는 두 클라이언트 등의 충돌하는 클라이언트 요청은 "최신 성공" 기준으로 해결됩니다. "Latest-WINS" 평가 시기는 S3 클라이언트가 작업을 시작할 때가 아니라 StorageGRID 시스템이 지정된 요청을 완료하는 시점을 기준으로 합니다.
- StorageGRID 버킷의 모든 오브젝트는 익명 사용자 또는 다른 계정에서 만든 오브젝트를 포함하여 버킷 소유자가 소유합니다.
- Swift를 통해 StorageGRID 시스템으로 수집된 데이터 오브젝트는 S3를 통해 액세스할 수 없습니다.

다음 표에서는 StorageGRID에서 S3 REST API 오브젝트 작업을 구현하는 방법을 설명합니다.

작동	구축
DeleteObject 를 클릭합니다	MFA(Multi-Factor Authentication) 및 응답 헤더입니다 `x-amz-mfa` 지원되지 않습니다. DeleteObject 요청을 처리할 때 StorageGRID는 저장된 모든 위치에서 개체의 모든 복사본을 즉시 제거하려고 시도합니다. 성공하면 StorageGRID는 즉시 클라이언트에 응답을 반환합니다. 위치를 일시적으로 사용할 수 없기 때문에 30초 이내에 모든 복사본을 제거할 수 없는 경우 StorageGRID는 제거할 복사본을 대기시킨 다음 클라이언트에 성공 여부를 표시합니다. 버전 관리 특정 버전을 제거하려면 요청자가 버킷 소유자여야 하며 를 사용해야 합니다 `versionId` 하위 리소스. 이 하위 리소스를 사용하면 버전이 영구적으로 삭제됩니다. 를 누릅니다 `versionId` 삭제 마커인 응답 헤더에 해당합니다 `x-amz-delete-marker` 가 로 설정된 상태로 반환됩니다 `true`. • 를 사용하지 않고 개체를 삭제한 경우 `versionId` 버전 지원 버킷의 하위 리소스에서는 삭제 마커가 생성됩니다. 를 클릭합니다 `versionId` 삭제 마커는 를 사용하여 반환됩니다 `x-amz-version-id` 응답 헤더 및 `x-amz-delete-marker` 로 설정된 응답 헤더가 반환됩니다 `true`. • 를 사용하지 않고 개체를 삭제한 경우 `versionId` 버전 일시 중지된 버킷의 하위 리소스는 기존 'null' 버전 또는 'null' 삭제 표식을 영구적으로 삭제하고 새 'null' 삭제 표식을 생성합니다. 를 클릭합니다 `x-amz-delete-marker` 로 설정된 응답 헤더가 반환됩니다 `true`. • 참고 *: 경우에 따라 객체에 대해 여러 개의 삭제 마커가 존재할 수 있습니다. 을 참조하십시오 "S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다" 거버넌스 모드에서 오브젝트 버전을 삭제하는 방법을 알아보십시오.

작동	구축
DeleteObjects 를 클릭합니다 (이전에 이름이 여러 개체 삭제)	MFA(Multi-Factor Authentication) 및 응답 헤더입니다 `x-amz-mfa` 지원되지 않습니다. 동일한 요청 메시지에서 여러 객체를 삭제할 수 있습니다. 을 참조하십시오 "S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다" 거버넌스 모드에서 오브젝트 버전을 삭제하는 방법을 알아보십시오.
DeleteObjectTagging 을 선택합니다	를 사용합니다 `tagging` 개체에서 모든 태그를 제거하는 하위 리소스입니다. 버전 관리 를 누릅니다 `versionId` 쿼리 매개 변수가 요청에 지정되지 않았습니다. 이 작업은 버전이 지정된 버킷에 있는 개체의 최신 버전에서 모든 태그를 삭제합니다. 개체의 현재 버전이 삭제 표식이면 와 함께 "MethodNotAllowed" 상태가 반환됩니다 `x-amz-delete-marker` 응답 헤더가 로 설정되었습니다 `true`.
GetObject 를 참조하십시오	"GetObject 를 참조하십시오"
GetObjectAcl	계정에 필요한 액세스 자격 증명이 제공된 경우 이 작업은 개체 소유자의 ID, DisplayName 및 사용 권한과 함께 긍정적인 응답을 반환합니다. 이는 소유자가 개체에 대한 모든 액세스 권한을 가지고 있음을 나타냅니다.
GetObjectLegalHold 를 참조하십시오	"S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다"
GetObjectRetention을 참조하십시오	"S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다"
GetObjectTagging	를 사용합니다 `tagging` 개체의 모든 태그를 반환하는 하위 리소스입니다. 버전 관리 를 누릅니다 `versionId` 쿼리 매개 변수가 요청에 지정되지 않았습니다. 이 작업은 버전 관리되는 버킷에서 가장 최신 버전의 개체에 있는 모든 태그를 반환합니다. 개체의 현재 버전이 삭제 표식이면 와 함께 "MethodNotAllowed" 상태가 반환됩니다 `x-amz-delete-marker` 응답 헤더가 로 설정되었습니다 `true`.
HeadObject 를 선택합니다	"HeadObject 를 선택합니다"
RestoreObject 를 선택합니다	"RestoreObject 를 선택합니다"
PutObject 를 선택합니다	"PutObject 를 선택합니다"

작동	구축
CopyObject 를 선택합니다 (이전에 명명된 Put Object - Copy)	"CopyObject 를 선택합니다"
PutObjectLegalHold를 선택합니다	"S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다"
PutObjectRetention	"S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다"
PutObjectTagging	를 사용합니다 `tagging` 기존 개체에 태그 집합을 추가하는 하위 리소스입니다. 개체 태그 제한 새 개체를 업로드할 때 태그를 추가하거나 기존 개체에 태그를 추가할 수 있습니다. StorageGRID 및 Amazon S3 모두 각 오브젝트에 대해 최대 10개의 태그를 지원합니다. 개체와 관련된 태그에는 고유한 태그 키가 있어야 합니다. 태그 키의 길이는 최대 128자의 유니코드 문자이고 태그 값의 길이는 최대 256자의 유니코드 문자일 수 있습니다. 키와 값은 대/소문자를 구분합니다. 태그 업데이트 및 수집 동작 PutObjectTagging을 사용하여 개체의 태그를 업데이트하는 경우 StorageGRID는 개체를 다시 수집하지 않습니다. 즉, 일치하는 ILM 규칙에 지정된 Ingest 동작 옵션이 사용되지 않습니다. ILM이 정상적인 백그라운드 ILM 프로세스에 의해 다시 평가될 때 업데이트로 인해 트리거되는 개체 배치에 대한 모든 변경 사항이 발생합니다. 즉, ILM 규칙이 수집 동작에 Strict 옵션을 사용하는 경우 필요한 개체 배치를 만들 수 없는 경우(예: 새로 필요한 위치를 사용할 수 없음) 작업이 수행되지 않습니다. 업데이트된 오브젝트는 필요한 배치가 가능할 때까지 현재 위치를 유지합니다. 충돌 해결 동일한 키에 쓰는 두 클라이언트 등의 충돌하는 클라이언트 요청은 "최신 성공" 기준으로 해결됩니다. "Latest-WINS" 평가 시기는 S3 클라이언트가 작업을 시작할 때가 아니라 StorageGRID 시스템이 지정된 요청을 완료하는 시점을 기준으로 합니다. 버전 관리 를 누릅니다 `versionId` 쿼리 매개 변수가 요청에 지정되지 않았습니다. 작업에서 버전 관리되는 버킷의 가장 최근 개체 버전에 태그를 추가합니다. 개체의 현재 버전이 삭제 표식이면 와 함께 "MethodNotAllowed" 상태가 반환됩니다 `x-amz-delete-marker` 응답 헤더가 로 설정되었습니다 `true`.
SelectObjectContent 를 선택합니다	"SelectObjectContent 를 선택합니다"

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

목록에 없는 항목은 지원되지 않습니다.

구문은 을 참조하십시오 "SelectObjectContent 를 선택합니다". S3 Select에 대한 자세한 내용은 를 참조하십시오 "S3 Select용 AWS 문서".

S3 Select가 활성화된 테넌트 계정만 SelectObjectContent 쿼리를 실행할 수 있습니다. 를 참조하십시오 "S3 Select 사용에 대한 고려 사항 및 요구 사항".

절을 참조하십시오

- 목록을 선택합니다
- FROM 절입니다
- WHERE 절
- Limit 절

데이터 유형

- 불입니다
- 정수
- 문자열
- 부동
- 십진수, 숫자
- 타임 스탬프입니다

연산자

논리 연산자

- 및
- 아닙니다
- 또는

비교 연산자

- 를 누릅니다
- 를 누릅니다
- lt;=.(&L
- GT;=.(&G
- =

- =
- 를 누릅니다
- !=
- 사이
- 인치

패턴 일치 연산자

- 좋아요
- _
- %

단일 작업자

- NULL입니다
- NULL이 아닙니다

수학 연산자

- 를 누릅니다
- -
- *
- /
- %

StorageGRID는 Amazon S3 Select 운영자 우선권을 따릅니다.

집계 함수

- 평균()
- 개수(*)
- 최대()
- 최소()
- 합계()

조건부 함수

- 케이스
- 합체
- 노LIF

변환 함수

- 캐스트(지원되는 데이터 형식용)

날짜 함수

- date_add
- Date_DIFF(날짜/시간
- 압축 풀기
- to_string(대상 문자열)
- 를 _TIMESTAMP로 설정합니다
- UTCNOW

문자열 함수

- char_length, character_length
- 낮음
- 부분 문자열
- 잘라내기
- 위쪽

서버측 암호화를 사용합니다

서버측 암호화를 통해 유휴 개체 데이터를 보호할 수 있습니다. StorageGRID는 개체를 쓸 때 데이터를 암호화하고 개체에 액세스할 때 데이터를 해독합니다.

서버측 암호화를 사용하려면 암호화 키가 관리되는 방식에 따라 상호 배타적인 두 가지 옵션 중 하나를 선택할 수 있습니다.

- * SSE(StorageGRID 관리 키를 사용한 서버 측 암호화) *: S3 요청을 발행하여 오브젝트를 저장할 때 StorageGRID는 고유 키를 사용하여 오브젝트를 암호화합니다. S3 요청을 통해 오브젝트를 검색할 때 StorageGRID는 저장된 키를 사용하여 오브젝트를 해독합니다.
- * SSE-C(고객이 제공한 키를 사용한 서버측 암호화) *: S3 요청을 발행하여 오브젝트를 저장할 때 사용자는 자신만의 암호화 키를 제공합니다. 오브젝트를 검색할 때 요청의 일부로 동일한 암호화 키를 제공합니다. 두 암호화 키가 일치하면 해당 개체는 해독되고 개체 데이터는 반환됩니다.

  StorageGRID는 모든 개체 암호화 및 암호 해독 작업을 관리하지만 사용자가 제공하는 암호화 키를 관리해야 합니다.

  제공한 암호화 키는 저장되지 않습니다. 암호화 키를 분실하면 해당 개체가 손실됩니다.

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

SSE를 사용합니다

StorageGRID에서 관리하는 고유 키를 사용하여 개체를 암호화하려면 다음 요청 헤더를 사용합니다.

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

SSE 요청 헤더는 다음 오브젝트 작업에서 지원됩니다.

- "PutObject 를 선택합니다"
- "CopyObject 를 선택합니다"
- "CreateMultptUpload 를 클릭합니다"

SSE-C를 사용합니다

관리하는 고유 키로 개체를 암호화하려면 다음 세 가지 요청 헤더를 사용합니다.

요청 헤더	설명
`x-amz-server-side-encryption-customer-algorithm`	암호화 알고리즘을 지정합니다. 헤더 값은 이어야 합니다 `AES256`.
`x-amz-server-side-encryption-customer-key`	개체를 암호화하거나 해독하는 데 사용할 암호화 키를 지정합니다. 키의 값은 256비트 base64로 인코딩되어야 합니다.
`x-amz-server-side-encryption-customer-key-MD5`	RFC 1321에 따라 암호화 키의 MD5 다이제스트를 지정합니다. RFC 1321은 암호화 키가 오류 없이 전송되도록 하는 데 사용됩니다. MD5 다이제스트 값은 base64로 인코딩된 128비트여야 합니다.

SSE-C 요청 헤더는 다음 개체 작업에서 지원됩니다.

- "GetObject 를 참조하십시오"
- "HeadObject 를 선택합니다"
- "PutObject 를 선택합니다"
- "CopyObject 를 선택합니다"
- "CreateMultptUpload 를 클릭합니다"
- "업로드 파트"
- "업로드파트 복사"

고객이 제공한 키(SSE-C)와 함께 서버측 암호화 사용 시 고려 사항

SSE-C를 사용하기 전에 다음 사항을 고려하십시오.

- https를 사용해야 합니다.

> StorageGRID는 SSE-C를 사용할 때 http를 통해 이루어진 요청을 거부합니다 보안을 위해 실수로 http를 사용하여 보낸 모든 키가 손상되지 않도록 고려해야 합니다. 키를 폐기하고 필요에 따라 회전합니다.

- 응답의 ETag는 객체 데이터의 MD5가 아닙니다.
- 암호화 키를 개체에 매핑하는 작업을 관리해야 합니다. StorageGRID는 암호화 키를 저장하지 않습니다. 각 개체에 대해 제공하는 암호화 키를 추적할 책임은 사용자에게 있습니다.
- 버킷을 버전 관리가 활성화된 경우 각 오브젝트 버전에는 고유한 암호화 키가 있어야 합니다. 각 개체 버전에

사용되는 암호화 키를 추적할 책임은 사용자에게 있습니다.

- 클라이언트 측에서 암호화 키를 관리하기 때문에 클라이언트 측에서 키 회전과 같은 추가 보호 수단을 관리해야 합니다.

> 제공한 암호화 키는 저장되지 않습니다. 암호화 키를 분실하면 해당 개체가 손실됩니다.

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

관련 정보

"Amazon S3 사용자 가이드: SSE-C(고객 제공 키)와 함께 서버측 암호화 사용"

CopyObject 를 선택합니다

S3 CopyObject 요청을 사용하여 이미 S3에 저장된 개체의 복사본을 만들 수 있습니다. CopyObject 작업은 GetObject 를 수행한 다음 PutObject 를 수행하는 작업과 같습니다.

충돌 해결

동일한 키에 쓰는 두 클라이언트 등의 충돌하는 클라이언트 요청은 "최신 성공" 기준으로 해결됩니다. "Latest-WINS" 평가 시기는 S3 클라이언트가 작업을 시작할 때가 아니라 StorageGRID 시스템이 지정된 요청을 완료하는 시점을 기준으로 합니다.

개체 크기

단일 PutObject 작업의 maximum_recommended_size는 5GiB(5,368,709,120바이트)입니다. 오브젝트가 5GiB보다 큰 경우 를 사용합니다 "멀티파트 업로드" 대신

단일 PutObject 작업의 maximum_supported_size는 5TiB(5,497,558,138,880바이트)입니다.

> StorageGRID 11.6 이하에서 업그레이드한 경우 5GiB를 초과하는 객체를 업로드하려고 하면 S3 PUT 개체 크기가 너무 큼 경고가 트리거됩니다. StorageGRID 11.7 또는 11.8을 새로 설치한 경우 경고가 트리거되지 않습니다. 하지만 StorageGRID의 향후 릴리즈에서는 AWS S3 표준에 맞춰 5GiB보다 큰 오브젝트 업로드를 지원하지 않습니다.

사용자 메타데이터의 **UTF-8** 문자

요청에 사용자 정의 메타데이터의 키 이름이나 값에 UTF-8 값이 포함되어 있으면 StorageGRID 동작이 정의되지 않습니다.

StorageGRID는 사용자 정의 메타데이터의 키 이름이나 값에 포함된 이스케이프된 UTF-8 문자를 구문 분석하거나 해석하지 않습니다. 이스케이프된 UTF-8 문자는 ASCII 문자로 처리됩니다.

- 사용자 정의 메타데이터에 이스케이프된 UTF-8 문자가 포함된 경우 요청이 성공합니다.
- StorageGRID는 을 반환하지 않습니다 `x-amz-missing-meta` 머리글 키 이름이나 값의 해석된 값에 인쇄할 수 없는 문자가 포함된 경우.

지원되는 요청 헤더입니다

지원되는 요청 헤더는 다음과 같습니다.

- `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-`사용자 정의 메타데이터를 포함하는 이름-값 쌍이 뒤에 옵니다
- `x-amz-metadata-directive`기본값은 입니다 `COPY`, 개체 및 관련 메타데이터를 복사할 수 있습니다.

  지정할 수 있습니다 `REPLACE` 오브젝트를 복사할 때 기존 메타데이터를 덮어쓰거나 오브젝트 메타데이터를 업데이트합니다.

- `x-amz-storage-class`
- `x-amz-tagging-directive`기본값은 입니다 `COPY`, 개체 및 모든 태그를 복사할 수 있습니다.

  지정할 수 있습니다 `REPLACE` 개체를 복사할 때 기존 태그를 덮어쓰거나 태그를 업데이트합니다.

- S3 오브젝트 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 버킷 기본 보존 설정을 사용하여 오브젝트 버전 모드와 보존 기간을 계산합니다. 을 참조하십시오 "S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다".

- 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`

    을 참조하십시오 서버측 암호화에 대한 요청 헤더

지원되지 않는 요청 헤더입니다

다음 요청 헤더는 지원되지 않습니다.

- `Cache-Control`

- `Content-Disposition`
- `Content-Encoding`
- `Content-Language`
- `Expires`
- `x-amz-website-redirect-location`

스토리지 클래스 옵션

를 클릭합니다 `x-amz-storage-class` 요청 헤더가 지원되며 일치하는 ILM 규칙이 이중 커밋 또는 밸런스를 사용하는 경우 StorageGRID에서 생성하는 객체 복제본 수에 영향을 줍니다 "수집 옵션".

- `STANDARD`

  (기본값) ILM 규칙이 이중 커밋 옵션을 사용하거나 균형 옵션이 중간 복사본 생성으로 돌아올 때 이중 커밋 수집 작업을 지정합니다.

- `REDUCED_REDUNDANCY`

  ILM 규칙이 이중 커밋 옵션을 사용하거나 균형 옵션이 중간 복사본 생성으로 돌아올 때 단일 커밋 수집 작업을 지정합니다.

  > S3 오브젝트 잠금이 설정된 버킷으로 오브젝트를 밀어넣는 경우, 를 참조하십시오 `REDUCED_REDUNDANCY` 옵션이 무시됩니다. 오브젝트를 레거시 준수 버킷으로 인스팅하는 경우, 를 참조하십시오 `REDUCED_REDUNDANCY` 옵션을 사용하면 오류가 반환됩니다. StorageGRID은 규정 준수 요구 사항이 충족될 수 있도록 항상 이중 커밋 수집을 수행합니다.

CopyObject에서 **x-amz-copy-source** 사용

소스 버킷과 키가 에 지정된 경우 `x-amz-copy-source` 헤더 는 대상 버킷 및 키와 다르며 소스 오브젝트 데이터의 복제본이 대상에 기록됩니다.

소스 및 대상이 일치하면, 및 입니다 `x-amz-metadata-directive` 머리글은 로 지정됩니다 `REPLACE`오브젝트의 메타데이터는 요청에 제공된 메타데이터 값으로 업데이트됩니다. 이 경우 StorageGRID는 오브젝트를 다시 수집하지 않습니다. 여기에는 두 가지 중요한 결과가 있습니다.

- 기존 개체를 현재 위치에서 암호화하거나 기존 개체의 암호화를 변경하는 데 CopyObject 를 사용할 수 없습니다. 를 공급하는 경우 `x-amz-server-side-encryption` 머리글 또는 을 선택합니다 `x-amz-server-side-encryption-customer-algorithm` header, StorageGRID가 요청을 거부하고 반환합니다 `XNotImplemented`.
- 일치하는 ILM 규칙에 지정된 Ingest 동작 옵션은 사용되지 않습니다. ILM이 정상적인 백그라운드 ILM 프로세스에 의해 다시 평가될 때 업데이트로 인해 트리거되는 개체 배치에 대한 모든 변경 사항이 발생합니다.

  즉, ILM 규칙이 수집 동작에 Strict 옵션을 사용하는 경우 필요한 개체 배치를 만들 수 없는 경우(예: 새로 필요한 위치를 사용할 수 없음) 작업이 수행되지 않습니다. 업데이트된 오브젝트는 필요한 배치가 가능할 때까지 현재 위치를 유지합니다.

서버측 암호화에 대한 요청 헤더

있다면 "서버측 암호화를 사용합니다"에서 제공하는 요청 헤더는 소스 개체가 암호화되었는지 여부와 대상 개체를 암호화할지 여부에 따라 달라집니다.

- 소스 객체가 SSE-C(고객 제공 키)를 사용하여 암호화되는 경우 CopyObject 요청에 다음 세 개의 헤더를 포함해야 객체를 해독한 후 복사할 수 있습니다.
  - `x-amz-copy-source-server-side-encryption-customer-algorithm`을 지정합니다 `AES256`.
  - `x-amz-copy-source-server-side-encryption-customer-key`: 소스 객체를 만들 때 제공한 암호화 키를 지정합니다.
  - `x-amz-copy-source-server-side-encryption-customer-key-MD5`: 소스 개체를 만들 때 제공한 MD5 다이제스트를 지정합니다.
- 제공 및 관리하는 고유 키를 사용하여 대상 개체(복사본)를 암호화하려면 다음 세 개의 머리글을 포함합니다.
  - `x-amz-server-side-encryption-customer-algorithm`을 지정합니다 `AES256`.
  - `x-amz-server-side-encryption-customer-key`: 대상 오브젝트의 새 암호화 키를 지정합니다.
  - `x-amz-server-side-encryption-customer-key-MD5`: 새 암호화 키의 MD5 다이제스트를 지정합니다.

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

- SSE(StorageGRID)에서 관리하는 고유 키를 사용하여 대상 개체(복사본)를 암호화하려는 경우 CopyObject 요청에 다음 헤더를 포함합니다.
  - `x-amz-server-side-encryption`

> 를 클릭합니다 `server-side-encryption` 개체의 값을 업데이트할 수 없습니다. 대신 새로 복사본을 만듭니다 `server-side-encryption` 값 사용 `x-amz-metadata-directive`: `REPLACE`.

버전 관리

소스 버킷의 버전이 있는 경우 를 사용할 수 있습니다 `x-amz-copy-source` Header - 개체의 최신 버전을 복사합니다. 특정 버전의 개체를 복사하려면 을 사용하여 복사할 버전을 명시적으로 지정해야 합니다 `versionId` 하위 리소스. 대상 버킷의 버전이 지정된 경우 생성된 버전이 에서 반환됩니다 `x-amz-version-id` 응답 헤더. 타겟 버킷에 대한 버전 관리가 일시 중지된 경우 `x-amz-version-id` "null" 값을 반환합니다.

GetObject 를 참조하십시오

S3 GetObject 요청을 사용하여 S3 버킷에서 객체를 검색할 수 있습니다.

GetObject 및 **multipart** 개체

를 사용할 수 있습니다 `partNumber` multipart 또는 segmented object의 특정 부분을 검색하기 위한 request 매개 변수입니다. 를 클릭합니다 `x-amz-mp-parts-count` Response 요소는 개체에 있는 파트 수를 나타냅니다.

을 설정할 수 있습니다 `partNumber` 분할/다중 파트 오브젝트 및 비분할/비다중 파트 오브젝트 모두에 대해 1로 설정하지만, 는 1로 설정됩니다 `x-amz-mp-parts-count` 응답 요소는 분할된 개체 또는 다중 파트 개체에 대해서만 반환됩니다.

사용자 메타데이터의 **UTF-8** 문자

StorageGRID는 사용자 정의 메타데이터에서 이스케이프된 UTF-8 문자를 구문 분석하거나 해석하지 않습니다. 사용자 정의 메타데이터에서 이스케이프된 UTF-8 문자가 있는 개체에 대한 요청을 가져오지 않습니다 `x-amz-missing-meta` 머리글 키 이름이나 값에 인쇄할 수 없는 문자가 포함된 경우.

지원되지 않는 요청 헤더입니다

다음 요청 헤더는 지원되지 않으며 반환됩니다 `XNotImplemented`:

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

버전 관리

가 있는 경우 `versionId` 하위 리소스가 지정되지 않았습니다. 작업이 버전 관리되는 버킷에서 개체의 최신 버전을 가져옵니다. 객체의 현재 버전이 삭제 표식인 경우, 와 함께 "찾을 수 없음" 상태가 반환됩니다 `x-amz-delete-marker` 응답 헤더가 로 설정되었습니다 `true`.

고객이 제공한 암호화 키**(SSE-C)**를 사용하여 서버측 암호화를 위한 요청 헤더

사용자가 제공한 고유 키로 개체를 암호화한 경우 머리글 3개를 모두 사용합니다.

- `x-amz-server-side-encryption-customer-algorithm`을 지정합니다 `AES256`.
- `x-amz-server-side-encryption-customer-key`: 오브젝트의 암호화 키를 지정합니다.
- `x-amz-server-side-encryption-customer-key-MD5`: 오브젝트의 암호화 키에 대한 MD5 다이제스트를 지정합니다.

> 제공한 암호화 키는 저장되지 않습니다. 암호화 키를 분실하면 해당 개체가 손실됩니다. 고객이 제공한 키를 사용하여 오브젝트 데이터를 보호하기 전에 의 고려 사항을 검토하십시오 "서버측 암호화를 사용합니다".

클라우드 스토리지 풀 객체에 대한 **GetObject**의 동작입니다

개체가 에 저장된 경우 "클라우드 스토리지 풀"GetObject 요청의 동작은 개체의 상태에 따라 다릅니다. 을 참조하십시오 "HeadObject 를 선택합니다" 를 참조하십시오.

> 오브젝트가 클라우드 스토리지 풀에 저장되고 하나 이상의 오브젝트 복제본이 그리드에 있는 경우 GetObject 요청은 클라우드 스토리지 풀에서 검색하기 전에 그리드에서 데이터 검색을 시도합니다.

개체의 상태입니다	**GetObject** 의 동작입니다
StorageGRID로 수집되었지만 아직 ILM에 의해 평가되지 않은 오브젝트 또는 기존 스토리지 풀에 저장된 오브젝트 또는 삭제 코딩 사용	`200 OK` 개체의 복사본이 검색됩니다.

개체의 상태입니다	**GetObject** 의 동작입니다
Cloud Storage Pool의 개체이지만 아직 검색할 수 없는 상태로 전환되지 않았습니다	`200 OK` 개체의 복사본이 검색됩니다.
개체가 검색할 수 없는 상태로 전환되었습니다	`403 Forbidden, InvalidObjectState` 를 사용합니다 "RestoreObject 를 선택합니다" 객체를 복구할 수 있는 상태로 복구하도록 요청합니다.
복구할 수 없는 상태에서 복원 중인 개체입니다	`403 Forbidden, InvalidObjectState` RestoreObject 요청이 완료될 때까지 기다립니다.
객체가 클라우드 스토리지 풀에 완전히 복구되었습니다	`200 OK` 개체의 복사본이 검색됩니다.

클라우드 스토리지 풀에서 다중 또는 분할 오브젝트

여러 부분 개체를 업로드했거나 StorageGRID가 큰 개체를 세그먼트로 분할한 경우 StorageGRID는 개체의 부분 또는 세그먼트의 하위 집합을 샘플링하여 클라우드 저장소 풀에서 개체를 사용할 수 있는지 여부를 결정합니다. 경우에 따라 GetObject 요청이 잘못 반환될 수 있습니다 `200 OK` 개체의 일부 부분이 이미 복구할 수 없는 상태로 전환되었거나 개체의 일부 부분이 아직 복원되지 않은 경우

다음과 같은 경우:

- GetObject 요청에서 일부 데이터를 반환하지만 전송 도중에 중지됩니다.
- 이후의 GetObject 요청이 반환될 수 있습니다 `403 Forbidden`.

GetObject 및 교차 그리드 복제

를 사용하는 경우 "그리드 통합" 및 "교차 그리드 복제" 버킷에 대해 활성화된 경우 S3 클라이언트는 GetObject 요청을 실행하여 객체의 복제 상태를 확인할 수 있습니다. 응답에는 StorageGRID에만 해당하는 것이 포함됩니다 `x-ntap-sg-cgr-replication-status` 다음 값 중 하나를 갖는 응답 헤더:

그리드	복제 상태입니다
출처	• * 성공 *: 복제가 성공했습니다. • * 보류 중 *: 객체가 아직 복제되지 않았습니다. • * 실패 *: 영구적인 장애로 인해 복제에 실패했습니다. 사용자가 오류를 해결해야 합니다.
목적지	• replica *: 객체가 소스 그리드에서 복제되었습니다.

StorageGRID는 을 지원하지 않습니다 `x-amz-replication-status` 머리글.

HeadObject 를 선택합니다

S3 HeadObject 요청을 사용하여 개체 자체를 반환하지 않고 개체에서 메타데이터를 검색할 수 있습니다. 객체가 클라우드 스토리지 풀에 저장된 경우 HeadObject 를 사용하여 객체의 전환 상태를 확인할 수 있습니다.

HeadObject 및 **multipart** 개체

를 사용할 수 있습니다 `partNumber` multipart 또는 segmented 객체의 특정 부분에 대한 메타데이터를 검색하는 request 매개 변수입니다. 를 클릭합니다 `x-amz-mp-parts-count` Response 요소는 개체에 있는 파트 수를 나타냅니다.

을 설정할 수 있습니다 `partNumber` 분할/다중 파트 오브젝트 및 비분할/비다중 파트 오브젝트 모두에 대해 1로 설정하지만, 는 1로 설정됩니다 `x-amz-mp-parts-count` 응답 요소는 분할된 개체 또는 다중 파트 개체에 대해서만 반환됩니다.

사용자 메타데이터의 **UTF-8** 문자

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

지원되지 않는 요청 헤더입니다

다음 요청 헤더는 지원되지 않으며 반환됩니다 `XNotImplemented`:

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

버전 관리

가 있는 경우 `versionId` 하위 리소스가 지정되지 않았습니다. 작업이 버전 관리되는 버킷에서 개체의 최신 버전을 가져옵니다. 객체의 현재 버전이 삭제 표식인 경우, 와 함께 "찾을 수 없음" 상태가 반환됩니다 `x-amz-delete-marker` 응답 헤더가 로 설정되었습니다 `true`.

고객이 제공한 암호화 키**(SSE-C)**를 사용하여 서버측 암호화를 위한 요청 헤더

사용자가 제공한 고유 키로 개체를 암호화한 경우 이 헤더 3개를 모두 사용합니다.

- `x-amz-server-side-encryption-customer-algorithm`을 지정합니다 `AES256`.
- `x-amz-server-side-encryption-customer-key`: 오브젝트의 암호화 키를 지정합니다.
- `x-amz-server-side-encryption-customer-key-MD5`: 오브젝트의 암호화 키에 대한 MD5 다이제스트를 지정합니다.

제공한 암호화 키는 저장되지 않습니다. 암호화 키를 분실하면 해당 개체가 손실됩니다. 고객이 제공한 키를 사용하여 오브젝트 데이터를 보호하기 전에 의 고려 사항을 검토하십시오 "서버측 암호화를 사용합니다".

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

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

응답 헤더는 클라우드 스토리지 풀로 이동되는 오브젝트의 상태에 대한 정보를 제공하며, 선택적으로 검색할 수 없는 상태로 전환된 후 복구됩니다.

개체의 상태입니다	**HeadObject**에 대한 응답입니다
StorageGRID로 수집되었지만 아직 ILM에 의해 평가되지 않은 오브젝트 또는 기존 스토리지 풀에 저장된 오브젝트 또는 삭제 코딩 사용	`200 OK` (특별한 응답 헤더가 반환되지 않습니다.)

Cloud Storage Pool의 개체이지만 아직 검색할 수 없는 상태로 전환되지 않았습니다	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	``x-amz-restore: progress-request="false", expiration-date="토요일, 2030년 7월 23일 00:00:00 GMT"
	
	개체가 검색할 수 없는 상태로 전환될 때까지 의 값은 입니다 `expiry-date` 앞으로 어느 정도 먼 시간으로 설정됩니다. 정확한 전환 시간은 StorageGRID 시스템에 의해 제어되지 않습니다.

개체가 검색할 수 없는 상태로 전환되었지만 하나 이상의 복사본이 그리드에 있습니다	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	``x-amz-restore: progress-request="false", expiration-date="토요일, 2030년 7월 23일 00:00:00 GMT"
	
	의 값 `expiry-date` 앞으로 어느 정도 먼 시간으로 설정됩니다.
	
	• 참고 *: 그리드의 복제본을 사용할 수 없는 경우(예: 스토리지 노드가 다운된 경우) 를 실행해야 합니다 "RestoreObject 를 선택합니다" 오브젝트를 성공적으로 검색하기 전에 클라우드 스토리지 풀에서 복사본을 복구하도록 요청합니다.

개체가 검색할 수 없는 상태로 전환되었으며 그리드에 복사본이 없습니다	`200 OK`
	
	`x-amz-storage-class: GLACIER`

개체의 상태입니다	**HeadObject**에 대한 응답입니다

복구할 수 없는 상태에서 복원 중인 개체입니다	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	``x-amz-restore:progress-request="true"``

객체가 클라우드 스토리지 풀에 완전히 복구되었습니다	`200 OK`
	
	`x-amz-storage-class: GLACIER`
	
	'x-amz-restore: progress-request="false", expiration-date="2018년 7월 23일 토요일 00:00:00 GMT"
	
	를 클릭합니다 `expiry-date` 클라우드 스토리지 풀의 객체가 검색 불가능한 상태로 반환되는 시점을 나타냅니다.

Cloud Storage Pool에서 다중 또는 분할 오브젝트 지원

여러 부분 개체를 업로드했거나 StorageGRID가 큰 개체를 세그먼트로 분할한 경우 StorageGRID는 개체의 부분 또는 세그먼트의 하위 집합을 샘플링하여 클라우드 저장소 풀에서 개체를 사용할 수 있는지 여부를 결정합니다. 개체의 일부가 이미 검색할 수 없는 상태로 전환되었거나 개체의 일부 부분이 아직 복원되지 않은 경우 HeadObject 요청이 ' x-amz-restore: accessive-request="false "를 잘못 반환할 수도 있습니다.

HeadObject 및 교차 그리드 복제

를 사용하는 경우 "그리드 통합" 및 "교차 그리드 복제" 버킷에 대해 활성화된 경우 S3 클라이언트는 HeadObject 요청을 실행하여 객체의 복제 상태를 확인할 수 있습니다. 응답에는 StorageGRID에만 해당하는 것이 포함됩니다 `x-ntap-sg-cgr-replication-status` 다음 값 중 하나를 갖는 응답 헤더:

그리드	복제 상태입니다
출처	• * 성공 *: 복제가 성공했습니다. • * 보류 중 *: 객체가 아직 복제되지 않았습니다. • * 실패 *: 영구적인 장애로 인해 복제에 실패했습니다. 사용자가 오류를 해결해야 합니다.
목적지	• replica *: 객체가 소스 그리드에서 복제되었습니다.

StorageGRID는 을 지원하지 않습니다 `x-amz-replication-status` 머리글.

PutObject 를 선택합니다

S3 PutObject 요청을 사용하여 객체를 버킷에 추가할 수 있습니다.

충돌 해결

동일한 키에 쓰는 두 클라이언트 등의 충돌하는 클라이언트 요청은 "최신 성공" 기준으로 해결됩니다. "Latest-WINS" 평가 시기는 S3 클라이언트가 작업을 시작할 때가 아니라 StorageGRID 시스템이 지정된 요청을 완료하는 시점을 기준으로 합니다.

개체 크기

단일 PutObject 작업의 maximum_recommended_size는 5GiB(5,368,709,120바이트)입니다. 오브젝트가 5GiB보다 큰 경우 를 사용합니다 "멀티파트 업로드" 대신

단일 PutObject 작업의 maximum_supported_size는 5TiB(5,497,558,138,880바이트)입니다.

> StorageGRID 11.6 이하에서 업그레이드한 경우 5GiB를 초과하는 객체를 업로드하려고 하면 S3 PUT 개체 크기가 너무 큼 경고가 트리거됩니다. StorageGRID 11.7 또는 11.8을 새로 설치한 경우 경고가 트리거되지 않습니다. 하지만 StorageGRID의 향후 릴리즈에서는 AWS S3 표준에 맞춰 5GiB보다 큰 오브젝트 업로드를 지원하지 않습니다.

사용자 메타데이터 크기입니다

Amazon S3는 각 PUT 요청 헤더 내의 사용자 정의 메타데이터 크기를 2KB로 제한합니다. StorageGRID는 사용자 메타데이터를 24KiB로 제한합니다. 사용자 정의 메타데이터의 크기는 각 키와 값의 UTF-8 인코딩에서 바이트 수의 합계를 구하여 측정됩니다.

사용자 메타데이터의 **UTF-8** 문자

요청에 사용자 정의 메타데이터의 키 이름이나 값에 UTF-8 값이 포함되어 있으면 StorageGRID 동작이 정의되지 않습니다.

StorageGRID는 사용자 정의 메타데이터의 키 이름이나 값에 포함된 이스케이프된 UTF-8 문자를 구문 분석하거나 해석하지 않습니다. 이스케이프된 UTF-8 문자는 ASCII 문자로 처리됩니다.

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

개체 태그 제한

새 개체를 업로드할 때 태그를 추가하거나 기존 개체에 태그를 추가할 수 있습니다. StorageGRID 및 Amazon S3 모두 각 오브젝트에 대해 최대 10개의 태그를 지원합니다. 개체와 관련된 태그에는 고유한 태그 키가 있어야 합니다. 태그 키의 길이는 최대 128자의 유니코드 문자이고 태그 값의 길이는 최대 256자의 유니코드 문자일 수 있습니다. 키와 값은 대/소문자를 구분합니다.

개체 소유권

StorageGRID에서는 소유자가 아닌 계정 또는 익명 사용자가 만든 개체를 포함하여 모든 개체가 버킷 소유자 계정에 의해 소유됩니다.

지원되는 요청 헤더입니다

지원되는 요청 헤더는 다음과 같습니다.

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

  를 지정할 때 `aws-chunked` 용 `Content-Encoding`StorageGRID는 다음 항목을 확인하지 않습니다.

  - StorageGRID에서 를 확인하지 않습니다 `chunk-signature` 청크 데이터를 기준으로 합니다.
  - StorageGRID는 사용자가 제공하는 값을 확인하지 않습니다 `x-amz-decoded-content-length` 반대.
- `Content-Language`
- `Content-Length`
- `Content-MD5`
- `Content-Type`
- `Expires`
- `Transfer-Encoding`

  청크된 전송 인코딩이 지원되는 경우 `aws-chunked` 페이로드 서명도 사용됩니다.

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

  사용자 정의 메타데이터에 대한 이름 값 쌍을 지정할 때 다음 일반 형식을 사용합니다.

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

  ILM 규칙의 참조 시간으로 * 사용자 정의 생성 시간 * 옵션을 사용하려면 을 사용해야 합니다 `creation-time` 오브젝트를 만들 때 기록하는 메타데이터의 이름입니다. 예를 들면 다음과 같습니다.

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

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

  > ⓘ ILM 규칙은 참조 시간에 * 사용자 정의 생성 시간 * 과 Balanced 또는 Strict 수집 옵션을 모두 사용할 수 없습니다. ILM 규칙을 만들면 오류가 반환됩니다.

- `x-amz-tagging`
- S3 오브젝트 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 버킷 기본 보존 설정을 사용하여 오브젝트 버전 모드와 보존 기간을 계산합니다. 을 참조하십시오 "S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다".

- 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`

    을 참조하십시오 서버측 암호화에 대한 요청 헤더

지원되지 않는 요청 헤더입니다

다음 요청 헤더는 지원되지 않습니다.

- 를 클릭합니다 `x-amz-acl` 요청 헤더가 지원되지 않습니다.
- 를 클릭합니다 `x-amz-website-redirect-location` 요청 헤더가 지원되지 않으며 반환됩니다 `XNotImplemented`.

스토리지 클래스 옵션

를 클릭합니다 `x-amz-storage-class` 요청 헤더가 지원됩니다. 에 대해 제출된 값입니다 `x-amz-storage-class` ILM을 통해 결정되는 StorageGRID 시스템에 저장된 개체의 영구 복사본 수가 아닌 수집 중에 StorageGRID이 오브젝트 데이터를 보호하는 방법에 영향을 미칩니다.

수집된 객체와 일치하는 ILM 규칙이 Strict 수집 옵션을 사용하는 경우 이 표시됩니다 `x-amz-storage-class` 머리글은 효과가 없습니다.

에 사용할 수 있는 값은 다음과 같습니다 `x-amz-storage-class`:

- `STANDARD` (기본값)
  - * 이중 커밋 *: ILM 규칙이 Ingest 동작에 대한 이중 커밋 옵션을 지정하는 경우, 개체가 수집되는 즉시 해당 개체의 두 번째 복사본이 생성되어 다른 스토리지 노드(이중 커밋)에 배포됩니다. ILM을 평가할 때 StorageGRID는 이러한 초기 중간 복사본이 규칙의 배치 지침을 충족하는지 여부를 결정합니다. 그렇지 않으면 새 오브젝트 복사본을 다른 위치에 만들어야 하고 초기 중간 복사본을 삭제해야 할 수 있습니다.
  - * 균형 *: ILM 규칙이 균형 옵션을 지정하고 StorageGRID이 규칙에 지정된 모든 복사본을 즉시 만들 수 없는 경우 StorageGRID은 다른 스토리지 노드에 두 개의 임시 복사본을 만듭니다.

    StorageGRID에서 ILM 규칙(동기식 배치)에 지정된 모든 오브젝트 복사본을 즉시 생성할 수 있는 경우 를 참조하십시오 `x-amz-storage-class` 머리글은 효과가 없습니다.
- `REDUCED_REDUNDANCY`
  - * 이중 커밋 *: ILM 규칙이 Ingest 동작에 대한 이중 커밋 옵션을 지정하는 경우 StorageGRID는 오브젝트가 수집될 때(단일 커밋) 단일 임시 복사본을 만듭니다.
  - * 균형 *: ILM 규칙이 균형 옵션을 지정하는 경우 StorageGRID은 시스템에서 규칙에 지정된 모든 사본을 즉시 만들 수 없는 경우에만 단일 중간 복사본을 만듭니다. StorageGRID에서 동기 배치를 수행할 수 있는 경우 이 머리글은 영향을 주지 않습니다. 를 클릭합니다 `REDUCED_REDUNDANCY` 옵션은 개체와 일치하는 ILM 규칙이 복제된 단일 복사본을 만들 때 가장 적합합니다. 이 경우 를 사용합니다 `REDUCED_REDUNDANCY` 모든 수집 작업에 대해 불필요한 오브젝트 복사본을 생성 및 삭제할 필요가 없습니다.

를 사용합니다 `REDUCED_REDUNDANCY` 다른 상황에서는 옵션을 사용하지 않는 것이 좋습니다. `REDUCED_REDUNDANCY` 수집 중에 오브젝트 데이터가 손실될 위험이 증가합니다. 예를 들어, ILM 평가가 발생하기 전에 실패한 스토리지 노드에 단일 복사본이 처음 저장되는 경우 데이터가 손실될 수 있습니다.

> 복제된 복사본이 항상 하나만 있으면 데이터가 영구적으로 손실될 위험이 있습니다. 복제된 객체 복제본이 하나만 있는 경우 스토리지 노드에 장애가 발생하거나 심각한 오류가 발생한 경우 해당 객체가 손실됩니다. 또한 업그레이드와 같은 유지보수 절차 중에는 개체에 대한 액세스가 일시적으로 중단됩니다.

지정 `REDUCED_REDUNDANCY` 오브젝트를 처음 인제스트할 때 생성되는 복사본 수에만 영향을 줍니다. 활성 ILM 정책에 따라 오브젝트를 평가할 때 생성되는 오브젝트 복사본 수에 영향을 미치지 않으며 StorageGRID 시스템에서 더 낮은 수준의 이중화로 데이터가 저장되지는 않습니다.

> S3 오브젝트 잠금이 설정된 버킷으로 오브젝트를 밀어넣는 경우, 를 참조하십시오 `REDUCED_REDUNDANCY` 옵션이 무시됩니다. 오브젝트를 레거시 준수 버킷으로 인스팅하는 경우, 를 참조하십시오 `REDUCED_REDUNDANCY` 옵션을 사용하면 오류가 반환됩니다. StorageGRID은 규정 준수 요구 사항이 충족될 수 있도록 항상 이중 커밋 수집을 수행합니다.

서버측 암호화에 대한 요청 헤더

다음 요청 헤더를 사용하여 서버측 암호화를 사용하여 개체를 암호화할 수 있습니다. SSE 및 SSE-C 옵션은 상호 배타적입니다.

- * SSE *: StorageGRID에서 관리하는 고유 키를 사용하여 오브젝트를 암호화하려면 다음 헤더를 사용하십시오.
  - `x-amz-server-side-encryption`
- * SSE-C *: 사용자가 제공 및 관리하는 고유 키로 객체를 암호화하려면 이 헤더 세 개를 모두 사용합니다.
  - `x-amz-server-side-encryption-customer-algorithm`을 지정합니다 `AES256`.
  - `x-amz-server-side-encryption-customer-key`새 오브젝트의 암호화 키를 지정합니다.
  - `x-amz-server-side-encryption-customer-key-MD5`: 새 개체의 암호화 키에 대한 MD5 다이제스트를 지정합니다.

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

> 개체가 SSE 또는 SSE-C로 암호화된 경우 버킷 수준 또는 그리드 수준 암호화 설정은 무시됩니다.

버전 관리

버켓에 대해 버전 관리가 활성화된 경우 고유한 가 사용됩니다 `versionId` 는 저장 중인 개체의 버전에 대해 자동으로 생성됩니다. 여기 `versionId` 를 사용하여 응답에서도 반환됩니다 `x-amz-version-id` 응답 헤더.

버전 관리가 일시 중단된 경우 개체 버전은 null로 저장됩니다 `versionId` null 버전이 이미 있는 경우 덮어쓰기가 됩니다.

승인 헤더의 서명 계산

를 사용할 때 `Authorization` 헤더 요청 인증, StorageGRID는 다음과 같은 방식으로 AWS와 다릅니다.

- StorageGRID은 필요하지 않습니다 `host` 에 포함할 헤더 `CanonicalHeaders`.
- StorageGRID은 필요하지 않습니다 `Content-Type` 에 포함될 수 있습니다 `CanonicalHeaders`.
- StorageGRID은 필요하지 않습니다 `x-amz-*` 에 포함할 헤더 `CanonicalHeaders`.

> 일반적인 모범 사례로서 항상 이 머리글을 안에 포함합니다 `CanonicalHeaders` 그러나 이러한 헤더를 제외하는 경우 StorageGRID에서 오류를 반환하지 않습니다.

자세한 내용은 을 참조하십시오 "승인 헤더에 대한 서명 계산:단일 청크(AWS 서명 버전 4)로 페이로드 전송".

관련 정보

"ILM을 사용하여 개체를 관리합니다"

RestoreObject 를 선택합니다

S3 RestoreObject 요청을 사용하여 클라우드 스토리지 풀에 저장된 개체를 복원할 수 있습니다.

지원되는 요청 유형입니다

StorageGRID에서는 객체를 복원하기 위한 RestoreObject 요청만 지원합니다. 는 지원하지 않습니다 `SELECT` 복원 유형. 반품 요청을 선택합니다 `XNotImplemented`.

버전 관리

필요에 따라 를 지정합니다 `versionId` 버전 관리되는 버킷에서 특정 버전의 오브젝트 복원 를 지정하지 않은 경우 `versionId`, 개체의 최신 버전이 복원됩니다

클라우드 스토리지 풀 객체에서 **RestoreObject**의 동작입니다

개체가 에 저장된 경우 "클라우드 스토리지 풀"RestoreObject 요청에는 개체의 상태에 따라 다음과 같은 동작이 있습니다. 을 참조하십시오 "HeadObject 를 선택합니다" 를 참조하십시오.

> 객체가 클라우드 스토리지 풀에 저장되어 있고 하나 이상의 객체 복제본도 그리드에 있는 경우 RestoreObject 요청을 실행하여 객체를 복구할 필요가 없습니다. 대신 GetObject 요청을 사용하여 로컬 복사본을 직접 검색할 수 있습니다.

개체의 상태입니다	**RestoreObject**의 동작입니다
StorageGRID로 수집되었지만 ILM에서 아직 평가되지 않은 오브젝트 또는 클라우드 스토리지 풀에 없는 오브젝트	`403 Forbidden, InvalidObjectState`
Cloud Storage Pool의 개체이지만 아직 검색할 수 없는 상태로 전환되지 않았습니다	`200 OK` 변경 사항이 없습니다. • 참고 *: 개체가 검색할 수 없는 상태로 전환되기 전에는 개체를 변경할 수 없습니다 `expiry-date`.

개체의 상태입니다	**RestoreObject**의 동작입니다
개체가 검색할 수 없는 상태로 전환되었습니다	`202 Accepted` 요청 본문에서 지정한 일 수에 대해 검색할 수 있는 객체 복제본을 클라우드 스토리지 풀에 복구합니다. 이 기간이 끝나면 객체는 복구할 수 없는 상태로 돌아갑니다. 필요에 따라 를 사용합니다 `Tier` 복원 작업을 완료하는 데 걸리는 시간을 결정하는 요청 요소입니다 (`Expedited`, `Standard`, 또는 `Bulk`)를 클릭합니다. 를 지정하지 않은 경우 `Tier`, `Standard` 계층이 사용됩니다. • 중요 *: 오브젝트가 S3 Glacier Deep Archive로 전환된 경우 또는 Cloud Storage Pool에서 Azure Blob 스토리지를 사용하는 경우 를 사용하여 복원할 수 없습니다 `Expedited` 계층. 다음 오류가 반환됩니다 `403 Forbidden`, `InvalidTier`: `Retrieval option is not supported by this storage class.`
복구할 수 없는 상태에서 복원 중인 개체입니다	`409 Conflict`, `RestoreAlreadyInProgress`
객체가 클라우드 스토리지 풀에 완전히 복구되었습니다	`200 OK` • 참고: * 개체가 검색 가능한 상태로 복원되면 이를 변경할 수 있습니다 `expiry-date` 에 대한 새 값으로 RestoreObject 요청을 다시 실행합니다 `Days`. 복원 날짜는 요청 시간을 기준으로 업데이트됩니다.

SelectObjectContent 를 선택합니다

S3 SelectObjectContent 요청을 사용하여 간단한 SQL 문을 기반으로 S3 개체의 내용을 필터링할 수 있습니다.

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

시작하기 전에

- 테넌트 계정에 S3 Select 권한이 있습니다.
- 있습니다 `s3:GetObject` 쿼리할 객체에 대한 권한입니다.
- 쿼리할 객체는 다음 형식 중 하나여야 합니다.
  - CSV *. GZIP 또는 BZIP2 보관 파일로 압축하거나 그대로 사용할 수 있습니다.
  - * 파케 *. Parquet 객체에 대한 추가 요구 사항:
    - S3 Select는 GZIP 또는 Snappy를 사용한 컬럼 압축만 지원합니다. S3 Select는 Parquet 오브젝트에 대한 전체 오브젝트 압축을 지원하지 않습니다.
    - S3 Select는 Parquet 출력을 지원하지 않습니다. 출력 형식을 CSV 또는 JSON으로 지정해야 합니다.
    - 압축되지 않은 최대 행 그룹 크기는 512MB입니다.
    - 개체의 스키마에 지정된 데이터 형식을 사용해야 합니다.
    - 간격, JSON, 목록, 시간 또는 UUID 논리적 유형은 사용할 수 없습니다.

- SQL 식의 최대 길이는 256KB입니다.
- 입력 또는 결과에 있는 모든 레코드의 최대 길이는 1MiB입니다.

CSV 요청 구문 예

```
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>
```

쪽모이 세공 요청 구문 예

```
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 쿼리의 예

이 쿼리는 시/도 이름, 2010년 인구, 2015년 예상 인구, 미국 인구 조사 데이터의 변경 비율을 가져옵니다. 상태가 아닌 파일의 레코드는 무시됩니다.

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

질의할 파일의 처음 몇 줄, `SUB-EST2020_ALL.csv`다음과 같이 보십시오.

```
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 예(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 줄, `changes.csv`다음과 같이 보십시오.

```
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 예(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 파일의 처음 몇 줄인 changes.csv는 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

다중 파트 업로드 작업: 개요

이 섹션에서는 StorageGRID가 멀티파트 업로드 작업을 지원하는 방법에 대해 설명합니다.

다음 조건 및 참고 사항은 모든 다중 파트 업로드 작업에 적용됩니다.

- 해당 버킷에 대한 ListMultipartUploads 쿼리의 결과가 불완전한 결과를 반환할 수 있으므로 단일 버킷에 대한 동시 다중 파트 업로드가 1,000개를 초과해서는 안 됩니다.
- StorageGRID는 여러 파트에 대해 AWS 크기 제한을 적용합니다. S3 클라이언트는 다음 지침을 따라야 합니다.
  - 멀티파트 업로드의 각 파트는 5MiB(5,242,880바이트)와 5GiB(5,368,709,120바이트) 사이여야 합니다.
  - 마지막 부분은 5MiB(5,242,880바이트)보다 작을 수 있습니다.
  - 일반적으로 파트 크기는 가능한 한 커야합니다. 예를 들어, 100GiB 개체의 경우 5GiB의 파트 크기를 사용합니다. 각 파트는 고유한 개체로 간주되므로 큰 파트 크기를 사용하면 StorageGRID 메타데이터 오버헤드가 줄어듭니다.
  - 5GiB보다 작은 오브젝트의 경우 대신 비다중 파트 업로드를 사용하는 것이 좋습니다.
- ILM 규칙이 Balanced 또는 Strict를 사용하는 경우, ILM은 수집 시 멀티파트 개체의 각 부분과 다중 파트 업로드가 완료될 때 개체 전체에 대해 평가됩니다 "수집 옵션". 이 사항이 개체 및 파트 배치에 미치는 영향에 대해 알고 있어야 합니다.
  - S3 다중 파트 업로드가 진행되는 동안 ILM이 변경되면 다중 파트 업로드가 완료될 때 개체의 일부 부분이 현재 ILM 요구사항을 충족하지 못할 수 있습니다. 올바르게 배치되지 않은 모든 부품은 ILM 재평가를 위해 대기열에 추가되고 나중에 올바른 위치로 이동됩니다.
  - 파트에 대한 ILM을 평가할 때 StorageGRID은 개체의 크기가 아닌 파트 크기를 필터링합니다. 즉, 개체의

일부를 개체에 대한 ILM 요구 사항을 전체가 충족하지 않는 위치에 저장할 수 있습니다. 예를 들어, 모든 작은 오브젝트가 DC2에 저장되지만 10GB 이상의 오브젝트는 모두 DC1에 저장되도록 규칙이 지정된 경우 10부분 다중 부분 업로드의 각 1GB 부분은 인제스트 시 DC2에 저장됩니다. 그러나 개체 전체에 대해 ILM을 평가하면 개체의 모든 부분이 DC1로 이동됩니다.

- 모든 멀티 파트 업로드 작업은 StorageGRID를 지원합니다 "일관성 값".
- 필요에 따라 를 사용할 수 있습니다 "서버 측 암호화" 멀티 파트 업로드가 가능합니다. SSE(StorageGRID 관리 키 사용 시 서버 측 암호화)를 사용하려면 를 포함합니다 `x-amz-server-side-encryption` CreateMultipartUpload 요청의 요청 헤더만. SSE-C(고객 제공 키를 사용한 서버측 암호화)를 사용하려면 CreateMultipartUpload 요청과 이후의 각 UploadPart 요청에서 동일한 3개의 암호화 키 요청 헤더를 지정합니다.

작동	구축
AbortMultipartUpload 를 클릭합니다	모든 Amazon S3 REST API 동작으로 구현됩니다. 예고 없이 변경될 수 있습니다.
CompleteMultipartUpload를 클릭합니다	을 참조하십시오 "CompleteMultipartUpload를 클릭합니다"
CreateMultptUpload 를 클릭합니다 (이전에 명명된 다중 파트 업로드 시작)	을 참조하십시오 "CreateMultptUpload 를 클릭합니다"
ListMultipartUploads 를 참조하십시오	을 참조하십시오 "ListMultipartUploads 를 참조하십시오"
목록 파트	모든 Amazon S3 REST API 동작으로 구현됩니다. 예고 없이 변경될 수 있습니다.
업로드 파트	을 참조하십시오 "업로드 파트"
업로드파트 복사	을 참조하십시오 "업로드파트 복사"

CompleteMultipartUpload를 클릭합니다

CompleteMultipartUpload 작업은 이전에 업로드한 부품을 조립하여 객체의 다중 부분 업로드를 완료합니다.

충돌 해결

동일한 키에 쓰는 두 클라이언트 등의 충돌하는 클라이언트 요청은 "최신 성공" 기준으로 해결됩니다. "Latest-WINS" 평가 시기는 S3 클라이언트가 작업을 시작할 때가 아니라 StorageGRID 시스템이 지정된 요청을 완료하는 시점을 기준으로 합니다.

요청 헤더

를 클릭합니다 `x-amz-storage-class` 요청 헤더가 지원되며 일치하는 ILM 규칙이 이중 커밋 또는 밸런스를 지정할 경우 StorageGRID에서 생성하는 객체 복제본 수에 영향을 줍니다 "수집 옵션".

- `STANDARD`

(기본값) ILM 규칙이 이중 커밋 옵션을 사용하거나 균형 옵션이 중간 복사본 생성으로 돌아올 때 이중 커밋 수집 작업을 지정합니다.

- `REDUCED_REDUNDANCY`

  ILM 규칙이 이중 커밋 옵션을 사용하거나 균형 옵션이 중간 복사본 생성으로 돌아올 때 단일 커밋 수집 작업을 지정합니다.

  > S3 오브젝트 잠금이 설정된 버킷으로 오브젝트를 밀어넣는 경우, 를 참조하십시오 `REDUCED_REDUNDANCY` 옵션이 무시됩니다. 오브젝트를 레거시 준수 버킷으로 인스팅하는 경우, 를 참조하십시오 `REDUCED_REDUNDANCY` 옵션을 사용하면 오류가 반환됩니다. StorageGRID은 규정 준수 요구 사항이 충족될 수 있도록 항상 이중 커밋 수집을 수행합니다.

> 15일 이내에 여러 부분 업로드가 완료되지 않으면 작업이 비활성으로 표시되고 모든 관련 데이터가 시스템에서 삭제됩니다.

> 를 클릭합니다 `ETag` 반환된 값은 MD5 합계가 아니라 의 Amazon S3 API 구현을 따릅니다 `ETag` 다중 파트 개체에 대한 값입니다.

버전 관리

이 작업은 여러 부분 업로드를 완료합니다. 버켓에 대해 버전 관리가 활성화된 경우 다중 파트 업로드가 완료된 후 개체 버전이 생성됩니다.

버켓에 대해 버전 관리가 활성화된 경우 고유한 가 사용됩니다 `versionId` 는 저장 중인 개체의 버전에 대해 자동으로 생성됩니다. 여기 `versionId` 를 사용하여 응답에서도 반환됩니다 `x-amz-version-id` 응답 헤더.

버전 관리가 일시 중단된 경우 개체 버전은 null로 저장됩니다 `versionId` null 버전이 이미 있는 경우 덮어쓰기가 됩니다.

> 버켓에 대해 버전 관리가 활성화된 경우, 같은 개체 키에서 동시 다중 파트 업로드가 완료된 경우에도 다중 파트 업로드를 완료하면 항상 새 버전이 생성됩니다. 버켓에 대해 버전 관리를 사용하지 않으면 다중 파트 업로드를 시작한 다음 다른 다중 파트 업로드를 시작하여 동일한 개체 키에서 먼저 완료할 수 있습니다. 비버전 버킷에서는 마지막으로 완료한 다중 파트 업로드가 우선 적용됩니다.

복제, 알림 또는 메타데이터 알림에 실패했습니다

플랫폼 서비스에 대해 다중 파트 업로드가 발생하는 버킷이 구성된 경우 연결된 복제 또는 알림 작업이 실패한 경우에도 다중 파트 업로드가 성공합니다.

이 경우 SMTT(Grid Manager on Total Events)에서 경보가 발생합니다. 마지막 이벤트 메시지에는 알림이 실패한 마지막 객체에 대한 "bucket-nameobject 키에 대한 알림을 게시하지 못했습니다."라는 메시지가 표시됩니다. (이 메시지를 보려면 * nodes * > **Storage Node* * > * Events * 를 선택합니다. 테이블 상단의 마지막 이벤트 보기) 이벤트 메시지는 에도 나열됩니다 `/var/local/log/bycast-err.log`.

테넌트는 개체의 메타데이터 또는 태그를 업데이트하여 실패한 복제 또는 알림을 트리거할 수 있습니다. 테넌트는 불필요한 변경을 방지하기 위해 기존 값을 다시 제출할 수 있습니다.

CreateMultptUpload 를 클릭합니다

CreateMultipartUpload(이전에 이름이 Multipart Upload 시작) 작업은 개체에 대한 다중 부분 업로드를 시작하고 업로드 ID를 반환합니다.

를 클릭합니다 `x-amz-storage-class` 요청 헤더가 지원됩니다. 에 대해 제출된 값입니다 `x-amz-storage-class` ILM을 통해 결정되는 StorageGRID 시스템에 저장된 개체의 영구 복사본 수가 아닌 수집 중에 StorageGRID이 오브젝트 데이터를 보호하는 방법에 영향을 미칩니다.

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

에 사용할 수 있는 값은 다음과 같습니다 `x-amz-storage-class`:

- `STANDARD` (기본값)
  - * Dual Commit *: ILM 규칙이 Dual Commit Ingest 옵션을 지정하는 경우 오브젝트가 수집되는 즉시 해당 오브젝트의 두 번째 복사본이 생성되어 다른 스토리지 노드(Dual Commit)로 배포됩니다. ILM을 평가할 때 StorageGRID는 이러한 초기 중간 복사본이 규칙의 배치 지침을 충족하는지 여부를 결정합니다. 그렇지 않으면 새 오브젝트 복사본을 다른 위치에 만들어야 하고 초기 중간 복사본을 삭제해야 할 수 있습니다.
  - * 균형 *: ILM 규칙이 균형 옵션을 지정하고 StorageGRID이 규칙에 지정된 모든 복사본을 즉시 만들 수 없는 경우 StorageGRID은 다른 스토리지 노드에 두 개의 임시 복사본을 만듭니다.

    StorageGRID에서 ILM 규칙(동기식 배치)에 지정된 모든 오브젝트 복사본을 즉시 생성할 수 있는 경우 를 참조하십시오 `x-amz-storage-class` 머리글은 효과가 없습니다.

- `REDUCED_REDUNDANCY`
  - * Dual Commit *: ILM 규칙이 Dual Commit 옵션을 지정하는 경우 StorageGRID는 개체가 수집될 때(단일 커밋) 하나의 중간 복사본을 생성합니다.
  - * 균형 *: ILM 규칙이 균형 옵션을 지정하는 경우 StorageGRID은 시스템에서 규칙에 지정된 모든 사본을 즉시 만들 수 없는 경우에만 단일 중간 복사본을 만듭니다. StorageGRID에서 동기 배치를 수행할 수 있는 경우 이 머리글은 영향을 주지 않습니다. 를 클릭합니다 `REDUCED_REDUNDANCY` 옵션은 개체와 일치하는 ILM 규칙이 복제된 단일 복사본을 만들 때 가장 적합합니다. 이 경우 를 사용합니다 `REDUCED_REDUNDANCY` 모든 수집 작업에 대해 불필요한 오브젝트 복사본을 생성 및 삭제할 필요가 없습니다.

  를 사용합니다 `REDUCED_REDUNDANCY` 다른 상황에서는 옵션을 사용하지 않는 것이 좋습니다. `REDUCED_REDUNDANCY` 수집 중에 오브젝트 데이터가 손실될 위험이 증가합니다. 예를 들어, ILM 평가가 발생하기 전에 실패한 스토리지 노드에 단일 복사본이 처음 저장되는 경우 데이터가 손실될 수 있습니다.

> 복제된 복사본이 항상 하나만 있으면 데이터가 영구적으로 손실될 위험이 있습니다. 복제된 객체 복제본이 하나만 있는 경우 스토리지 노드에 장애가 발생하거나 심각한 오류가 발생한 경우 해당 객체가 손실됩니다. 또한 업그레이드와 같은 유지보수 절차 중에는 개체에 대한 액세스가 일시적으로 중단됩니다.

지정 `REDUCED_REDUNDANCY` 오브젝트를 처음 인제스트할 때 생성되는 복사본 수에만 영향을 줍니다. 활성 ILM 정책에 따라 오브젝트를 평가할 때 생성되는 오브젝트 복사본 수에 영향을 미치지 않으며 StorageGRID 시스템에서 더 낮은 수준의 이중화로 데이터가 저장되지는 않습니다.

S3 오브젝트 잠금이 설정된 버킷으로 오브젝트를 밀어넣는 경우, 를 참조하십시오 `REDUCED_REDUNDANCY` 옵션이 무시됩니다. 오브젝트를 레거시 준수 버킷으로 인스팅하는 경우, 를 참조하십시오 `REDUCED_REDUNDANCY` 옵션을 사용하면 오류가 반환됩니다. StorageGRID은 규정 준수 요구 사항이 충족될 수 있도록 항상 이중 커밋 수집을 수행합니다.

지원되는 요청 헤더는 다음과 같습니다.

- `Content-Type`
- `x-amz-meta-`사용자 정의 메타데이터를 포함하는 이름-값 쌍이 뒤에 옵니다

  사용자 정의 메타데이터에 대한 이름 값 쌍을 지정할 때 다음 일반 형식을 사용합니다.

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

  ILM 규칙의 참조 시간으로 * 사용자 정의 생성 시간 * 옵션을 사용하려면 을 사용해야 합니다 `creation-time` 오브젝트를 만들 때 기록하는 메타데이터의 이름입니다. 예를 들면 다음과 같습니다.

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

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

  추가 중 `creation-time` 레거시 규정 준수 기능이 설정된 버킷에 오브젝트를 추가할 경우 사용자 정의 메타데이터가 허용되지 않습니다. 오류가 반환됩니다.

- S3 오브젝트 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 버킷 기본 보존 설정을 사용하여 개체 버전 보존 기간을 계산합니다.

  "S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다"

- 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`

    서버측 암호화에 대한 요청 헤더

StorageGRID에서 UTF-8 문자를 처리하는 방법에 대한 자세한 내용은 을 참조하십시오 "PutObject 를 선택합니다".

서버측 암호화에 대한 요청 헤더

다음 요청 헤더를 사용하여 서버측 암호화를 사용하여 다중 파트 개체를 암호화할 수 있습니다. SSE 및 SSE-C 옵션은 상호 배타적입니다.

- * sse *: StorageGRID에서 관리하는 고유 키로 개체를 암호화하려면 CreateMultipartUpload 요청에서 다음 헤더를 사용합니다. UploadPart 요청에는 이 헤더를 지정하지 마십시오.
  - `x-amz-server-side-encryption`
- * SSE-C *: 제공 및 관리하는 고유 키로 개체를 암호화하려면 CreateMultipartUpload 요청(및 이후의 각 UploadPart 요청)에 이 헤더 세 개를 모두 사용하십시오.
  - `x-amz-server-side-encryption-customer-algorithm`을 지정합니다 `AES256`.
  - `x-amz-server-side-encryption-customer-key`새 오브젝트의 암호화 키를 지정합니다.
  - `x-amz-server-side-encryption-customer-key-MD5`: 새 개체의 암호화 키에 대한 MD5 다이제스트를 지정합니다.

제공한 암호화 키는 저장되지 않습니다. 암호화 키를 분실하면 해당 개체가 손실됩니다. 고객이 제공한 키를 사용하여 오브젝트 데이터를 보호하기 전에 의 고려 사항을 검토하십시오 "서버 측 암호화 사용".

지원되지 않는 요청 헤더입니다

다음 요청 헤더는 지원되지 않으며 반환됩니다 `XNotImplemented`

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

버전 관리

멀티파트 업로드는 업로드 시작, 리스팅 업로드, 파트 업로드, 업로드된 파트 조립 및 업로드 완료를 위한 별도의 작업으로 구성됩니다. CompleteMultipartUpload 작업이 수행될 때 객체가 생성되고 해당되는 경우 버전이 지정됩니다.

ListMultipartUploads 를 참조하십시오

ListMultipartUploads 작업은 버킷에 대해 진행 중인 다중 파트 업로드를 나열합니다.

지원되는 요청 매개 변수는 다음과 같습니다.

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

- `Date`
- `Authorization`

버전 관리

멀티파트 업로드는 업로드 시작, 리스팅 업로드, 파트 업로드, 업로드된 파트 조립 및 업로드 완료를 위한 별도의 작업으로 구성됩니다. CompleteMultipartUpload 작업이 수행될 때 객체가 생성되고 해당되는 경우 버전이 지정됩니다.

업로드 파트

UploadPart 작업은 객체에 대한 다중 부분 업로드의 파트를 업로드합니다.

지원되는 요청 헤더입니다

지원되는 요청 헤더는 다음과 같습니다.

- `Content-Length`
- `Content-MD5`

서버측 암호화에 대한 요청 헤더

CreateMultipartUpload 요청에 대해 SSE-C 암호화를 지정한 경우 각 UploadPart 요청에 다음 요청 머리글도 포함해야 합니다.

- `x-amz-server-side-encryption-customer-algorithm`을 지정합니다 `AES256`.
- `x-amz-server-side-encryption-customer-key`: CreateMultipartUpload 요청에서 제공한 것과 동일한 암호화 키를 지정합니다.
- `x-amz-server-side-encryption-customer-key-MD5`: CreateMultipartUpload 요청에 제공한 것과 동일한 MD5 다이제스트를 지정합니다.

> 제공한 암호화 키는 저장되지 않습니다. 암호화 키를 분실하면 해당 개체가 손실됩니다. 고객이 제공한 키를 사용하여 오브젝트 데이터를 보호하기 전에 의 고려 사항을 검토하십시오 "서버측 암호화를 사용합니다".

버전 관리

멀티파트 업로드는 업로드 시작, 리스팅 업로드, 파트 업로드, 업로드된 파트 조립 및 업로드 완료를 위한 별도의 작업으로 구성됩니다. CompleteMultipartUpload 작업이 수행될 때 객체가 생성되고 해당되는 경우 버전이 지정됩니다.

업로드파트 복사

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

UploadPartCopy 작업은 모든 Amazon S3 REST API 동작으로 구현됩니다. 예고 없이 변경될 수 있습니다.

이 요청은 에 지정된 오브젝트 데이터를 읽고 씁니다 `x-amz-copy-source-range` StorageGRID 시스템 내에서

지원되는 요청 헤더는 다음과 같습니다.

- `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`

서버측 암호화에 대한 요청 헤더

CreateMultipartUpload 요청에 대해 SSE-C 암호화를 지정한 경우 각 UploadPartCopy 요청에 다음 요청 머리글도 포함해야 합니다.

- `x-amz-server-side-encryption-customer-algorithm`을 지정합니다 `AES256`.
- `x-amz-server-side-encryption-customer-key`: CreateMultipartUpload 요청에서 제공한 것과 동일한 암호화 키를 지정합니다.
- `x-amz-server-side-encryption-customer-key-MD5`: CreateMultipartUpload 요청에 제공한 것과 동일한 MD5 다이제스트를 지정합니다.

소스 객체가 SSE-C(고객 제공 키)를 사용하여 암호화되는 경우 UploadPartCopy 요청에 다음 세 개의 헤더를 포함해야 객체를 해독한 후 복사할 수 있습니다.

- `x-amz-copy-source-server-side-encryption-customer-algorithm`을 지정합니다 `AES256`.
- `x-amz-copy-source-server-side-encryption-customer-key`: 소스 객체를 만들 때 제공한 암호화 키를 지정합니다.
- `x-amz-copy-source-server-side-encryption-customer-key-MD5`: 소스 개체를 만들 때 제공한 MD5 다이제스트를 지정합니다.

> 제공한 암호화 키는 저장되지 않습니다. 암호화 키를 분실하면 해당 개체가 손실됩니다. 고객이 제공한 키를 사용하여 오브젝트 데이터를 보호하기 전에 의 고려 사항을 검토하십시오 "서버측 암호화를 사용합니다".

버전 관리

멀티파트 업로드는 업로드 시작, 리스팅 업로드, 파트 업로드, 업로드된 파트 조립 및 업로드 완료를 위한 별도의 작업으로 구성됩니다. CompleteMultipartUpload 작업이 수행될 때 객체가 생성되고 해당되는 경우 버전이 지정됩니다.

오류 응답

StorageGRID 시스템은 적용되는 모든 표준 S3 REST API 오류 응답을 지원합니다. 또한 StorageGRID 구현에는 여러 개의 사용자 지정 응답이 추가됩니다.

지원되는 **S3 API** 오류 코드입니다

이름	**HTTP** 상태입니다
액세스가 거부되었습니다	403 사용 금지
배다이제스트	400 잘못된 요청

이름	**HTTP** 상태입니다
BuckketAlreadyExists를 참조하십시오	409 충돌
BuckketNotEmpty	409 충돌
IncompleteBody	400 잘못된 요청
내부 오류입니다	500 내부 서버 오류입니다
InvalidAccessKeyId 입니다	403 사용 금지
InvalidArgument 를 선택합니다	400 잘못된 요청
InvalidBuckName입니다	400 잘못된 요청
InvalidBuckketState입니다	409 충돌
InvalidDigest 를 선택합니다	400 잘못된 요청
InvalidEncryptionAlgorithmError 가 발생합니다	400 잘못된 요청
InvalidPart 를 선택합니다	400 잘못된 요청
InvalidPartOrder를 선택합니다	400 잘못된 요청
InvalidRange 를 선택합니다	416 요청된 범위가 충분하지 않습니다
InvalidRequest 입니다	400 잘못된 요청
InvalidStorageClass 의 값을 반환합니다	400 잘못된 요청
InvalidTag 를 선택합니다	400 잘못된 요청
InvalidURI입니다	400 잘못된 요청
키투롱	400 잘못된 요청
MalformedXML을 참조하십시오	400 잘못된 요청
MetadataTooLarge를 참조하십시오	400 잘못된 요청
MethodNotAllowed 를 참조하십시오	405 메서드를 사용할 수 없습니다

이름	**HTTP** 상태입니다
MissingContentLength를 참조하십시오	411 길이 필요
MissingRequestBodyError가 발생합니다	400 잘못된 요청
MissingSecurityHeader 를 참조하십시오	400 잘못된 요청
NoSuchBucket	404를 찾을 수 없습니다
NoSuchKey를 클릭합니다	404를 찾을 수 없습니다
NoSuchUpload 를 클릭합니다	404를 찾을 수 없습니다
구현되지 않았습니다	501 구현되지 않음
NoSuchBuckketPolicy를 참조하십시오	404를 찾을 수 없습니다
ObjectLockConfigurationNotFoundError 가 발생합니다	404를 찾을 수 없습니다
사전 조건에 실패했습니다	412 전제 조건 실패
RequestTimeTooSkewed 를 참조하십시오	403 사용 금지
서비스를 사용할 수 없습니다	503 서비스를 사용할 수 없습니다
SignatureDoesNotMatch 를 참조하십시오	403 사용 금지
투만이버킷	400 잘못된 요청
UserKeyMustBeSpecified 를 선택합니다	400 잘못된 요청

StorageGRID 사용자 지정 오류 코드

이름	설명	**HTTP** 상태입니다
XBuckketLifecycleNotAllowed를 참조하십시오	버킷 수명 주기 구성은 레거시 준수 버킷에서 허용되지 않습니다	400 잘못된 요청
XBuckketPolicyParseException 을 참조하십시오	수신된 버킷 정책 JSON을 구문 분석하지 못했습니다.	400 잘못된 요청
XComplianceConflictt	레거시 준수 설정으로 인해 작업이 거부되었습니다.	403 사용 금지

이름	설명	HTTP 상태입니다
XComplianceRedundancyForbidden을 선택합니다	레거시 준수 버킷에서는 감소된 중복성이 허용되지 않습니다	400 잘못된 요청
XMaxBuckketPolicyLengthExceeded 를 참조하십시오	정책이 허용되는 최대 버킷 정책 길이를 초과합니다.	400 잘못된 요청
XMissingInternalRequestHeader를 참조하십시오	내부 요청의 헤더가 누락되었습니다.	400 잘못된 요청
XNoSuchBuckketCompliance	지정된 버킷에 레거시 준법 기능이 설정되어 있지 않습니다.	404를 찾을 수 없습니다
XNotAcceptable(X 허용 가능)	요청에 충족되지 않은 하나 이상의 수락 헤더가 있습니다.	406 허용되지 않습니다
XNotImplemented(XNotImplemented)	제공한 요청은 구현되지 않은 기능을 의미합니다.	501 구현되지 않음

StorageGRID 사용자 정의 작업

StorageGRID 사용자 정의 작업: 개요

StorageGRID 시스템은 S3 REST API에 추가된 사용자 지정 작업을 지원합니다.

다음 표에서는 StorageGRID에서 지원하는 사용자 지정 작업을 보여 줍니다.

작동	설명
"버킷 일관성 확보"	특정 버킷에 적용되는 일관성을 반환합니다.
"버킷 일관성을 유지합니다"	특정 버킷에 적용되는 일관성을 설정합니다.
"버킷 최종 액세스 시간 가져오기"	특정 버킷에 대해 마지막 액세스 시간 업데이트를 사용할 수 있는지 여부를 반환합니다.
"버킷 최종 접근 시간"	특정 버킷에 대한 마지막 액세스 시간 업데이트를 활성화하거나 비활성화할 수 있습니다.
"버킷 메타데이터 알림 구성을 삭제합니다"	특정 버킷과 연결된 메타데이터 알림 구성 XML을 삭제합니다.
"Bucket 메타데이터 알림 구성 가져오기"	특정 버킷과 연결된 메타데이터 알림 구성 XML을 반환합니다.

작동	설명
"Put Bucket 메타데이터 알림 구성"	버킷에 대한 메타데이터 알림 서비스를 구성합니다.
"스토리지 사용량을 가져옵니다"	계정과 연결된 각 버킷에서 사용 중인 총 저장소 양을 나타냅니다.
"사용되지 않음: 규정 준수 설정이 있는 CreateBucket"	더 이상 사용되지 않으며 지원되지 않음: Compliance를 사용하는 새 버킷을 더 이상 생성할 수 없습니다.
"사용되지 않음: 버킷 준수 가져오기"	더 이상 사용되지 않지만 지원됨: 기존 레거시 준수 버킷에 대해 현재 적용되는 규정 준수 설정을 반환합니다.
"사용되지 않음: 버킷 준수"	사용되지 않지만 지원됨: 기존 레거시 준수 버킷의 준수 설정을 수정할 수 있습니다.

버킷 일관성 확보

Get Bucket Consistency 요청을 사용하면 특정 버킷에 적용되는 일관성을 확인할 수 있습니다.

기본 정합성 보장은 새로 생성된 개체에 대해 쓰기 후 읽기를 보장하도록 설정됩니다.

이 작업을 완료하려면 S3:GetBucketConsistency 권한이 있거나 계정 루트여야 합니다.

요청 예

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

응답

응답 XML에서 `<Consistency>` 다음 값 중 하나를 반환합니다.

정합성	설명
모두	모든 노드가 데이터를 즉시 수신하거나 요청이 실패합니다.
강함 - 글로벌	모든 사이트에서 모든 클라이언트 요청에 대해 쓰기 후 읽기 정합성을 보장합니다.
강력한 사이트	사이트 내의 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다.

정합성	설명
읽기-후-새로-쓰기	(기본값) 새 객체에 대한 읽기 후 쓰기 정합성을 보장하고 객체 업데이트에 대한 최종 일관성을 제공합니다. 고가용성 및 데이터 보호 보장 제공 대부분의 경우에 권장됩니다.
사용 가능	새 개체 및 개체 업데이트 모두에 대한 최종 일관성을 제공합니다. S3 버킷의 경우 필요한 경우에만 사용하십시오(예: 거의 읽지 않는 로그 값이 포함된 버킷의 경우 또는 존재하지 않는 키의 헤드 또는 GET 작업의 경우). S3 FabricPool 버킷은 지원되지 않습니다.

응답 예

```
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 값"

버킷 일관성을 유지합니다

Put Bucket Consistency 요청을 사용하면 버킷에서 수행된 작업에 적용할 일관성을 지정할 수 있습니다.

기본 정합성 보장은 새로 생성된 개체에 대해 쓰기 후 읽기를 보장하도록 설정됩니다.

시작하기 전에

이 작업을 완료하려면 S3:PutBucketConsistency 권한이 있거나 계정 루트여야 합니다.

요청하십시오

를 클릭합니다 `x-ntap-sg-consistency` 매개 변수는 다음 값 중 하나를 포함해야 합니다.

정합성	설명
모두	모든 노드가 데이터를 즉시 수신하거나 요청이 실패합니다.

정합성	설명
강함 - 글로벌	모든 사이트에서 모든 클라이언트 요청에 대해 쓰기 후 읽기 정합성을 보장합니다.
강력한 사이트	사이트 내의 모든 클라이언트 요청에 대해 쓰기 후 읽기 일관성을 보장합니다.
읽기-후-새로-쓰기	(기본값) 새 객체에 대한 읽기 후 쓰기 정합성을 보장하고 객체 업데이트에 대한 최종 일관성을 제공합니다. 고가용성 및 데이터 보호 보장 제공 대부분의 경우에 권장됩니다.
사용 가능	새 개체 및 개체 업데이트 모두에 대한 최종 일관성을 제공합니다. S3 버킷의 경우 필요한 경우에만 사용하십시오(예: 거의 읽지 않는 로그 값이 포함된 버킷의 경우 또는 존재하지 않는 키의 헤드 또는 GET 작업의 경우). S3 FabricPool 버킷은 지원되지 않습니다.

- 참고: * 일반적으로 "Read-after-new-write" 일관성을 사용해야 합니다. 요청이 올바르게 작동하지 않는 경우 가능한 경우 응용 프로그램 클라이언트 동작을 변경합니다. 또는 각 API 요청의 일관성을 지정하도록 클라이언트를 구성합니다. 버킷 수준의 일관성을 마지막 수단으로 설정합니다.

요청 예

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

관련 정보

"일관성 값"

버킷 최종 액세스 시간 가져오기

[버킷 최종 액세스 시간 가져오기(Get Bucket Last Access Time) 요청 을 사용하면 개별 버킷에 대해 마지막 액세스 시간 업데이트가 활성화되거나 비활성화되었는지 확인할 수 있습니다.

이 작업을 완료하려면 S3:GetBucketLastAccessTime 권한이 있거나 계정 루트여야 합니다.

요청 예

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

응답 예

이 예에서는 버킷에 대해 마지막 액세스 시간 업데이트가 활성화되어 있음을 보여 줍니다.

```
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>
```

버킷 최종 접근 시간

Put Bucket Last Access Time 요청을 사용하면 개별 버킷에 대한 마지막 액세스 시간 업데이트를 활성화하거나 비활성화할 수 있습니다. 마지막 액세스 시간 업데이트를 비활성화하면 성능이 향상되고 버전 10.3.0 이상으로 생성된 모든 버킷의 기본 설정이 됩니다.

이 작업을 완료하려면 버킷에 대한 S3:PutBuckketLastAccessTime 권한이 있거나 계정 루트여야 합니다.

> StorageGRID 버전 10.3부터는 모든 새 버킷에 대해 마지막 액세스 시간에 대한 업데이트가 기본적으로 비활성화됩니다. 이전 버전의 StorageGRID를 사용하여 만든 버킷이 있고 새 기본 동작과 일치시키려면 이전의 각 버킷에 대해 마지막 액세스 시간 업데이트를 명시적으로 비활성화해야 합니다. Put Bucket 마지막 액세스 시간 요청을 사용하거나 Tenant Manager의 버킷에 대한 세부 정보 페이지에서 마지막 액세스 시간에 대한 업데이트를 활성화 또는 비활성화할 수 있습니다. 을 참조하십시오 "마지막 액세스 시간 업데이트를 사용하거나 사용하지 않도록 설정합니다".

버킷에 대해 마지막 액세스 시간 업데이트가 비활성화된 경우 버킷의 작업에 다음 동작이 적용됩니다.

- GetObject, GetObjectAcl, GetObjectTagging 및 HeadObject 요청은 마지막 액세스 시간을 업데이트하지 않습니다. ILM(정보 수명 주기 관리) 평가를 위해 객체가 대기열에 추가되지 않습니다.
- 메타데이터만 업데이트하는 CopyObject 및 PutObjectTagging 요청도 마지막 액세스 시간을 업데이트합니다. ILM 평가를 위해 오브젝트가 대기열에 추가됩니다.
- 소스 버킷에 대해 마지막 액세스 시간에 대한 업데이트를 사용할 수 없는 경우 CopyObject 요청이 소스 버킷의 마지막 액세스 시간을 업데이트하지 않습니다. 복사된 객체는 소스 버킷에 대한 ILM 평가를 위해 대기열에 추가되지 않습니다. 그러나 대상의 경우 CopyObject 요청은 항상 마지막 액세스 시간을 업데이트합니다. ILM 평가를 위해 개체의 복사본이 대기열에 추가됩니다.
- CompleteMultipartUpload 요청이 마지막 액세스 시간을 업데이트합니다. 완료된 객체가 ILM 평가를 위해 대기열에 추가됩니다.

예를 요청하십시오

이 예제에서는 버킷의 마지막 액세스 시간을 설정합니다.

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

이 예제에서는 버킷의 마지막 액세스 시간을 비활성화합니다.

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

버킷 메타데이터 알림 구성을 삭제합니다

Delete Bucket 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 예

이 예제에서는 버킷에 대한 검색 통합 서비스를 비활성화하는 방법을 보여 줍니다.

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

Bucket 메타데이터 알림 구성 가져오기

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

이 작업을 완료하려면 S3:GetBuckMetadataNotification 권한이 있거나 계정 루트여야 합니다.

요청 예

이 요청은 이름이 인 버킷에 대한 메타데이터 알림 구성을 검색합니다 `bucket`.

```
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 사용하면 버킷이 검색 통합을 위해 구성되는 방식을 결정할 수 있습니다. 즉, 인덱싱된 개체와 해당 개체 메타데이터가 전송되는 끝점을 확인할 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 적용되는 오브젝트와 StorageGRID가 오브젝트 메타데이터를 보내야 하는 대상을 지정합니다. 대상은 StorageGRID 끝점의 URN을 사용하여 지정해야 합니다.

이름	설명	필수 요소입니다
MetadataNotificationConfiguration을 참조하십시오	메타데이터 알림의 개체 및 대상을 지정하는 데 사용되는 규칙의 컨테이너 태그입니다. 하나 이상의 규칙 요소가 포함되어 있습니다.	예

이름	설명	필수 요소입니다
규칙	메타데이터를 지정된 인덱스에 추가해야 하는 개체를 식별하는 규칙의 컨테이너 태그입니다. 접두사가 겹치는 규칙은 거부됩니다. MetadataNotificationConfiguration 요소에 포함되어 있습니다.	예
ID입니다	규칙의 고유 식별자입니다. Rule 요소에 포함되어 있습니다.	아니요
상태	상태는 '활성화' 또는 '비활성화'가 될 수 있습니다. 비활성화된 규칙에 대해 어떠한 작업도 수행되지 않습니다. Rule 요소에 포함되어 있습니다.	예
접두어	접두사와 일치하는 개체는 규칙의 영향을 받으며 해당 메타데이터는 지정된 대상으로 전송됩니다. 모든 오브젝트를 일치시키려면 빈 접두사를 지정합니다. Rule 요소에 포함되어 있습니다.	예
목적지	규칙의 대상에 대한 컨테이너 태그입니다. Rule 요소에 포함되어 있습니다.	예

이름	설명	필수 요소입니다
urn	객체 메타데이터가 전송되는 대상의 urn입니다. 다음 속성을 가진 StorageGRID 끝점의 URN이어야 합니다. • `es` 세 번째 요소여야 합니다. • URN은 메타데이터가 저장된 인덱스 및 형식으로 양식에 끝나야 합니다 `domain-name/myindex/mytype`. 엔드포인트는 테넌트 관리자 또는 테넌트 관리 API를 사용하여 구성됩니다. 다음과 같은 형식을 취합니다. • `arn:aws:es:_region:account-ID_:domain/mydomain/myindex/mytype` • `urn:mysite:es:::mydomain/myindex/mytype` 구성 XML을 제출하기 전에 끝점을 구성해야 합니다. 그렇지 않으면 404 오류로 인해 구성이 실패합니다. urn은 Destination 요소에 포함됩니다.	예

응답 예

사이에 포함된 XML
`<MetadataNotificationConfiguration></MetadataNotificationConfiguration>` 태그는 버킷에 대해 검색 통합 끝점과의 통합이 어떻게 구성되어 있는지 보여줍니다. 이 예제에서는 개체 메타데이터가 라는 Elasticsearch 인덱스에 전송되고 있습니다 `current` 이름을 입력합니다 `2017` 라는 AWS 도메인에서 호스팅됩니다 `records`.

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

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

관련 정보

"테넌트 계정을 사용합니다"

Put Bucket 메타데이터 알림 구성

Put Bucket 메타데이터 알림 구성 요청을 사용하면 개별 버킷에 대한 검색 통합 서비스를 활성화할 수 있습니다. 요청 본문에 제공하는 메타데이터 알림 구성 XML은 대상 검색 인덱스에 메타데이터가 전송되는 개체를 지정합니다.

이 작업을 완료하려면 버킷에 대한 S3:PutBucketMetadataNotification 권한이 있거나 계정 루트여야 합니다.

요청하십시오

요청 본문에는 메타데이터 알림 구성이 포함되어야 합니다. 각 메타데이터 알림 구성에는 하나 이상의 규칙이 포함됩니다. 각 규칙은 적용되는 오브젝트와 StorageGRID가 오브젝트 메타데이터를 보내야 하는 대상을 지정합니다.

개체 이름의 접두어를 기준으로 개체를 필터링할 수 있습니다. 예를 들어, 접두사가 있는 개체에 대한 메타데이터를 보낼 수 있습니다 `/images` 목적지 하나와 접두사가 있는 오브젝트 `/videos` 다른 사람에게.

중복되는 접두사가 있는 구성은 유효하지 않으며 제출될 때 거부됩니다. 예를 들어, 접두사가 있는 개체에 대해 하나의 규칙이 포함된 구성입니다 `test` 접두사가 있는 개체에 대한 두 번째 규칙입니다 `test2` 허용되지 않습니다.

대상은 StorageGRID 끝점의 URN을 사용하여 지정해야 합니다. 메타데이터 알림 구성이 제출되거나 요청이 로 실패하는 경우 엔드포인트가 있어야 합니다 `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 표에서는 메타데이터 알림 구성 XML의 요소에 대해 설명합니다.

이름	설명	필수 요소입니다
MetadataNotificationConfiguration을 참조하십시오	메타데이터 알림의 개체 및 대상을 지정하는 데 사용되는 규칙의 컨테이너 태그입니다. 하나 이상의 규칙 요소가 포함되어 있습니다.	예
규칙	메타데이터를 지정된 인덱스에 추가해야 하는 개체를 식별하는 규칙의 컨테이너 태그입니다. 접두사가 겹치는 규칙은 거부됩니다. MetadataNotificationConfiguration 요소에 포함되어 있습니다.	예
ID입니다	규칙의 고유 식별자입니다. Rule 요소에 포함되어 있습니다.	아니요
상태	상태는 '활성화' 또는 '비활성화'가 될 수 있습니다. 비활성화된 규칙에 대해 어떠한 작업도 수행되지 않습니다. Rule 요소에 포함되어 있습니다.	예

이름	설명	필수 요소입니다
접두어	접두사와 일치하는 개체는 규칙의 영향을 받으며 해당 메타데이터는 지정된 대상으로 전송됩니다. 모든 오브젝트를 일치시키려면 빈 접두사를 지정합니다. Rule 요소에 포함되어 있습니다.	예
목적지	규칙의 대상에 대한 컨테이너 태그입니다. Rule 요소에 포함되어 있습니다.	예
urn	객체 메타데이터가 전송되는 대상의 urn입니다. 다음 속성을 가진 StorageGRID 끝점의 URN이어야 합니다. • `es` 세 번째 요소여야 합니다. • URN은 메타데이터가 저장된 인덱스 및 형식으로 양식에 끝나야 합니다 `domain-name/myindex/mytype`. 엔드포인트는 테넌트 관리자 또는 테넌트 관리 API를 사용하여 구성됩니다. 다음과 같은 형식을 취합니다. • `arn:aws:es:region:account-ID:domain/mydomain/myindex/mytype` • `urn:mysite:es:::mydomain/myindex/mytype` 구성 XML을 제출하기 전에 끝점을 구성해야 합니다. 그렇지 않으면 404 오류로 인해 구성이 실패합니다. urn은 Destination 요소에 포함됩니다.	예

예를 요청하십시오

이 예제에서는 버킷에 대한 검색 통합을 활성화하는 방법을 보여 줍니다. 이 예제에서 모든 오브젝트의 오브젝트 메타데이터는 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 예제에서는 접두사와 일치하는 개체의 개체 메타데이터를 보여 줍니다 `/images` 은(는) 한 대상으로 전송되지만 접두사와 일치하는 오브젝트의 오브젝트 메타데이터는 전송됩니다 `/videos` 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>
```

JSON이 검색 통합 서비스에 의해 생성되었습니다

버킷에 대한 검색 통합 서비스를 활성화하면 개체 메타데이터 또는 태그를 추가, 업데이트 또는 삭제할 때마다 JSON 문서가 생성되어 대상 끝점으로 전송됩니다.

이 예제에서는 키가 있는 개체가 생성될 수 있는 JSON의 예를 보여 줍니다 `SGWS/Tagging.txt` 이(가) 라는 이름의 버킷에 생성됩니다 `test`. 를 클릭합니다 `test` 버킷의 버전이 지정되지 않으므로 이(가) 이(가) 필요합니다 `versionId` 태그가 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 개체 메타데이터입니다

이 표에는 검색 통합이 활성화된 경우 대상 끝점으로 전송되는 JSON 문서에 포함된 모든 필드가 나열됩니다.

문서 이름에는 버킷 이름, 오브젝트 이름 및 버전 ID(있는 경우)가 포함됩니다.

유형	항목 이름	설명
버킷 및 오브젝트 정보	버킷	버킷의 이름입니다
버킷 및 오브젝트 정보	키	개체 키 이름입니다
버킷 및 오브젝트 정보	버전 ID	오브젝트 버전, 버전 버킷 내 오브젝트
버킷 및 오브젝트 정보	지역	버킷 영역(예 `us-east-1`
시스템 메타데이터	크기	HTTP 클라이언트에 표시되는 개체 크기(바이트)입니다
시스템 메타데이터	MD5	개체 해시

유형	항목 이름	설명
사용자 메타데이터	메타데이터 *key:value*	객체에 대한 모든 사용자 메타데이터를 키 값 쌍으로 사용합니다
태그	태그 *key:value*	개체에 대해 정의된 모든 개체 태그를 키 값 쌍으로 사용합니다

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

관련 정보

"테넌트 계정을 사용합니다"

스토리지 사용 요청 가져오기

Get Storage Usage 요청은 계정이 사용하는 총 스토리지 양과 계정과 연결된 각 버킷에 대해 알려줍니다.

계정과 해당 버킷에서 사용하는 스토리지의 양은 을 사용하여 수정된 ListBucket 요청을 통해 얻을 수 있습니다 `x-ntap-sg-usage` 쿼리 매개 변수입니다. 시스템에서 처리하는 PUT 및 삭제 요청과는 별도로 버킷 스토리지 사용량을 추적합니다. 특히 시스템이 과부하 상태인 경우, 사용 값이 요청 처리를 기준으로 예상 값과 일치하기 전에 약간의 지연이 있을 수 있습니다.

기본적으로 StorageGRID는 강력한 글로벌 일관성을 사용하여 사용 정보 검색을 시도합니다. 강력한 글로벌 일관성을 달성할 수 없는 경우 StorageGRID는 강력한 사이트 일관성으로 사용 정보를 검색합니다.

이 작업을 완료하려면 S3:ListAllMyBucket 권한이 있거나 계정 루트여야 합니다.

요청 예

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

응답 예

이 예에서는 두 버킷에 4개의 오브젝트와 12바이트의 데이터가 있는 계정을 보여 줍니다. 각 버킷에는 2개의 오브젝트와 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 개체 버전은 에 기여합니다 `ObjectCount` 및 `DataBytes` 응답의 값입니다. Delete markers(마커 삭제)는 에 추가되지 않습니다 `ObjectCount` 합계.

관련 정보

"일관성 값"

레거시 규정 준수를 위한 더 이상 사용되지 않는 버킷 요청

레거시 규정 준수를 위한 더 이상 사용되지 않는 버킷 요청

StorageGRID S3 REST API를 사용하여 레거시 규정 준수 기능을 사용하여 생성된 버킷을 관리해야 할 수 있습니다.

규정 준수 기능이 사용되지 않습니다

이전 StorageGRID 버전에서 사용할 수 있었던 StorageGRID 규정 준수 기능은 더 이상 사용되지 않으며 S3 오브젝트 잠금으로 대체되었습니다.

이전에 글로벌 규정 준수 설정을 활성화한 경우 StorageGRID 11.6에서 전역 S3 개체 잠금 설정이 활성화됩니다. Compliance를 사용하도록 설정한 상태에서 새 버킷을 더 이상 생성할 수 없지만, 필요에 따라 StorageGRID S3 REST API를 사용하여 기존의 규정을 준수하는 버킷을 관리할 수 있습니다.

- "S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다"
- "ILM을 사용하여 개체를 관리합니다"
- "NetApp 기술 자료: StorageGRID 11.5에서 레거시 준수 버킷을 관리하는 방법"

더 이상 사용되지 않는 규정 준수 요청:

- "폐기됨 - 규정 준수를 위해 버킷 요청을 수정합니다"

  SGCompliance XML 요소는 사용되지 않습니다. 이전 버전에서는 이 StorageGRID 사용자 정의 요소를 PUT 버킷 요청의 선택적 XML 요청 본문에 포함하여 준수 버킷을 생성할 수 있었습니다.

- "사용되지 않음 - 버킷 규정 준수"

  Get Bucket 준수 요청은 더 이상 사용되지 않습니다. 그러나 이 요청을 계속 사용하여 기존 레거시 준수 버킷에 현재 적용되는 준수 설정을 확인할 수 있습니다.

- "사용되지 않음 - Put 버킷 준수"

  PUT 버킷 준수 요청은 더 이상 사용되지 않습니다. 그러나 이 요청을 계속 사용하여 기존 레거시 준수 버킷의 준수 설정을 수정할 수 있습니다. 예를 들어, 기존 버킷을 법적 보존 상태로 놓거나 보존 기간을 늘릴 수 있습니다.

사용되지 않음: 규정 준수를 위한 **CreateBucket** 요청 수정

SGCompliance XML 요소는 사용되지 않습니다. 이전에는 이 StorageGRID 사용자 지정 요소를 CreateBucket 요청의 선택적 XML 요청 본문에 포함시켜 준수 버킷을 만들 수 있었습니다.

> 이전 StorageGRID 버전에서 사용할 수 있었던 StorageGRID 규정 준수 기능은 더 이상 사용되지 않으며 S3 오브젝트 잠금으로 대체되었습니다. 자세한 내용은 다음을 참조하십시오.
>
> - "S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다"
> - "NetApp 기술 자료: StorageGRID 11.5에서 레거시 준수 버킷을 관리하는 방법"

Compliance가 설정된 새 버킷을 더 이상 생성할 수 없습니다. 규정 준수를 위해 CreateBucket 요청 수정을 사용하여 새 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.
```

사용되지 않음: 버킷 준수 요청 가져오기

Get Bucket 준수 요청은 더 이상 사용되지 않습니다. 그러나 이 요청을 계속 사용하여 기존

레거시 준수 버킷에 현재 적용되는 준수 설정을 확인할 수 있습니다.

> 이전 StorageGRID 버전에서 사용할 수 있었던 StorageGRID 규정 준수 기능은 더 이상 사용되지 않으며 S3 오브젝트 잠금으로 대체되었습니다. 자세한 내용은 다음을 참조하십시오.
>
> - "S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다"
> - "NetApp 기술 자료: StorageGRID 11.5에서 레거시 준수 버킷을 관리하는 방법"

이 작업을 완료하려면 S3:GetBucketCompliance 권한이 있거나 계정 루트여야 합니다.

요청 예

이 요청 예제를 통해 이름이 인 버킷의 준수 설정을 확인할 수 있습니다 `mybucket`.

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

응답 예

응답 XML에서 `<SGCompliance>` 에는 버킷에 적용되는 준수 설정이 나와 있습니다. 이 예제 응답에서는 오브젝트를 그리드에 인제스트하는 시점을 시작으로 각 오브젝트를 1년(525,600분)동안 보존할 버킷의 규정 준수 설정을 보여 줍니다. 현재 이 버킷에 대한 법적 보류가 없습니다. 각 개체는 1년 후에 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
RetenionPeriodMinutes(주기적 지연 시간)	이 버킷에 추가된 객체의 보존 기간(분)입니다. 보존 기간은 객체가 그리드에 수집될 때 시작됩니다.

이름	설명
LegalHold	• 참: 이 버킷은 현재 법적 증거 자료 보관 중입니다. 이 버킷의 오브젝트는 보존 기간이 만료된 경우에도 법적 보류가 해제될 때까지 삭제할 수 없습니다. • 거짓: 이 버킷은 현재 법적 증거 자료 보관 중이 아닙니다. 이 버킷의 오브젝트는 보존 기간이 만료되면 삭제할 수 있습니다.
자동 삭제	• 참: 버킷이 법적 보존 상태에 있지 않는 한, 보존 기간이 만료되면 이 버킷의 객체가 자동으로 삭제됩니다. • False: 보존 기간이 만료되면 이 버킷의 객체가 자동으로 삭제되지 않습니다. 이러한 개체를 삭제하려면 해당 개체를 수동으로 삭제해야 합니다.

오류 응답

버킷이 규정을 준수하도록 생성되지 않은 경우 응답에 대한 HTTP 상태 코드는 입니다 `404 Not Found`, 의 S3 오류 코드 포함 `XNoSuchBucketCompliance`.

폐기됨: 버킷 준수 요청을 넣으십시오

PUT 버킷 준수 요청은 더 이상 사용되지 않습니다. 그러나 이 요청을 계속 사용하여 기존 레거시 준수 버킷의 준수 설정을 수정할 수 있습니다. 예를 들어, 기존 버킷을 법적 보존 상태로 놓거나 보존 기간을 늘릴 수 있습니다.

> 이전 StorageGRID 버전에서 사용할 수 있었던 StorageGRID 규정 준수 기능은 더 이상 사용되지 않으며 S3 오브젝트 잠금으로 대체되었습니다. 자세한 내용은 다음을 참조하십시오.
>
> • "S3 REST API를 사용하여 S3 오브젝트 잠금을 구성합니다"
> • "NetApp 기술 자료: StorageGRID 11.5에서 레거시 준수 버킷을 관리하는 방법"

이 작업을 완료하려면 S3:PutBucketCompliance 권한이 있거나 계정 루트 권한이 있어야 합니다.

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

요청 예

이 예제 요청은 이름이 인 버킷의 준수 설정을 수정합니다 `mybucket`. 이 예제에서는 의 개체를 보여 줍니다 `mybucket` 이제 오브젝트를 그리드로 인제스트하는 시점을 시작으로 1년이 아닌 2년(1,051,200분) 동안 보존됩니다. 이 버킷에는 법적 구속이 없습니다. 각 개체는 2년 후에 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
RetenionPeriodMinutes(주기적 지연 시간)	이 버킷에 추가된 객체의 보존 기간(분)입니다. 보존 기간은 객체가 그리드에 수집될 때 시작됩니다. • 중요 * RetentionPeriodMinutes 에 새 값을 지정할 때는 버킷의 현재 보존 기간과 같거나 큰 값을 지정해야 합니다. 버킷의 보존 기간이 설정된 후에는 해당 값을 줄일 수 없으며 증가만 가능합니다.
LegalHold	• 참: 이 버킷은 현재 법적 증거 자료 보관 중입니다. 이 버킷의 오브젝트는 보존 기간이 만료된 경우에도 법적 보류가 해제될 때까지 삭제할 수 없습니다. • 거짓: 이 버킷은 현재 법적 증거 자료 보관 중이 아닙니다. 이 버킷의 오브젝트는 보존 기간이 만료되면 삭제할 수 있습니다.
자동 삭제	• 참: 버킷이 법적 보존 상태에 있지 않는 한, 보존 기간이 만료되면 이 버킷의 객체가 자동으로 삭제됩니다. • False: 보존 기간이 만료되면 이 버킷의 객체가 자동으로 삭제되지 않습니다. 이러한 개체를 삭제하려면 해당 개체를 수동으로 삭제해야 합니다.

규정 준수 설정에 대한 일관성

PUT 버킷 준수 요청으로 S3 버킷의 준수 설정을 업데이트하면 StorageGRID는 그리드 전체에서 버킷의 메타데이터를 업데이트하려고 시도합니다. 기본적으로 StorageGRID은 * 강력한 글로벌 * 일관성을 사용하여 버킷 메타데이터가 포함된 모든 데이터 센터 사이트와 모든 스토리지 노드에서 변경된 규정 준수 설정에 대해 쓰기 후 읽기 정합성을 보장합니다.

데이터 센터 사이트 또는 사이트의 여러 스토리지 노드를 사용할 수 없기 때문에 StorageGRID에서 * 강력한 글로벌 * 일관성을 달성할 수 없는 경우 응답에 대한 HTTP 상태 코드는 입니다 `503 Service Unavailable`.

이 응답을 받으면 그리드 관리자에게 문의하여 필요한 스토리지 서비스를 가능한 빨리 사용할 수 있도록 해야 합니다. 그리드 관리자가 각 사이트에서 스토리지 노드를 충분히 사용할 수 없는 경우 기술 지원 부서에서 * 강력한 사이트 * 일관성을 적용하여 실패한 요청을 다시 시도하도록 지시할 수 있습니다.

기술 지원 부서의 지시가 있거나 이 레벨을 사용할 때 발생할 수 있는 결과를 이해하지 않는 한, Put Bucket 준수를 위해 *Strong-site * 일관성을 강제로 적용하지 마십시오.

일관성이 * 강력한 사이트 * 로 감소하면 StorageGRID는 업데이트된 규정 준수 설정이 사이트 내 클라이언트 요청에 대해서만 쓰기 후 읽기 일관성을 유지할 수 있도록 보장합니다. 즉, 모든 사이트 및 스토리지 노드를 사용할 수 있을 때까지 StorageGRID 시스템에 이 버킷에 대한 여러 개의 일관되지 않은 설정이 일시적으로 있을 수 있습니다. 설정이 일치하지 않으면 예기치 않거나 원치 않는 동작이 발생할 수 있습니다. 예를 들어, 버킷을 법적 증거 자료 보관 중에 두고 일관성을 더 낮게 설정하면 일부 데이터 센터 사이트에서 버킷의 이전 규정 준수 설정(즉, 법적 증거 자료 보관)이 계속 적용될 수 있습니다. 따라서 보존 기간이 만료되면 사용자나 자동 삭제(활성화된 경우)에 의해 법적 보류라고 생각하는 개체가 삭제될 수 있습니다.

Strong-site * 일관성을 강제로 사용하려면 Put Bucket 준수 요청을 다시 발행하고 를 포함시킵니다 `Consistency-Control` HTTP 요청 헤더는 다음과 같습니다.

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

오류 응답

- 버킷이 규정을 준수하도록 생성되지 않은 경우 응답에 대한 HTTP 상태 코드는 입니다 `404 Not Found`.
- If(경우 `RetentionPeriodMinutes` 요청이 버킷의 현재 보존 기간보다 작은 경우 HTTP 상태 코드는 입니다 `400 Bad Request`.

관련 정보

"사용되지 않음: 규정 준수를 위해 버킷 요청 수정을 배치합니다"

버킷 및 그룹 액세스 정책

버킷 및 그룹 액세스 정책을 사용합니다

StorageGRID은 AWS(Amazon Web Services) 정책 언어를 사용하여 S3 테넌트가 해당 버킷 및 오브젝트 내의 버킷에 대한 액세스를 제어할 수 있도록 합니다. StorageGRID 시스템은 S3 REST API 정책 언어의 하위 집합을 구현합니다. S3 API에 대한 액세스 정책은 JSON으로 기록됩니다.

액세스 정책 개요

StorageGRID에서 지원하는 액세스 정책에는 두 가지 유형이 있습니다.

- *버킷 정책 * - GetBucketPolicy, PutBucketPolicy 및 DeleteBucketPolicy S3 API 작업을 사용하여 관리됩니다. 버킷 정책은 버킷에 첨부되므로 버킷 소유자 계정 또는 버킷에 대한 다른 계정 및 버킷에 있는 오브젝트에 대한 사용자의 액세스를 제어하도록 구성됩니다. 버킷 정책은 하나의 버킷과 여러 그룹에만 적용됩니다.
- 테넌트 관리자 또는 테넌트 관리 API를 사용하여 구성된 * 그룹 정책 * 입니다. 그룹 정책은 계정의 그룹에 연결되므로 해당 그룹이 해당 계정이 소유한 특정 리소스에 액세스할 수 있도록 구성됩니다. 그룹 정책은 하나의 그룹에만 적용되고 여러 버킷에 적용될 수 있습니다.

그룹 정책과 버킷 정책 간에는 우선 순위에 차이가 없습니다.

StorageGRID 버킷 및 그룹 정책은 아마존에서 정의한 특정 문법을 따릅니다. 각 정책 안에는 정책 문의 배열이 들어 있으며 각 문에는 다음 요소가 포함되어 있습니다.

- 정책 ID(SID)(선택 사항)
- 효과
- Principal/NotPrincipal입니다
- 리소스/NotResource입니다
- 작업/NotAction
- 조건(선택 사항)

정책 문은 이 구조를 사용하여 권한을 지정합니다. pert <effect> <principal>이(가) <condition>이(가) 적용될 때 <Resource>에서 <Action>을(를) 수행하도록 허용/거부합니다.

각 정책 요소는 특정 함수에 사용됩니다.

요소	설명
SID	SID 요소는 선택 사항입니다. SID는 사용자에 대한 설명으로만 제공됩니다. StorageGRID 시스템에서 저장하지만 해석되지 않습니다.
효과	Effect 요소를 사용하여 지정된 작업의 허용 여부를 설정합니다. 지원되는 작업 요소 키워드를 사용하여 버킷 또는 오브젝트에 대해 허용(또는 거부)하는 작업을 식별해야 합니다.
Principal/NotPrincipal입니다	사용자, 그룹 및 계정이 특정 리소스에 액세스하고 특정 작업을 수행하도록 허용할 수 있습니다. 요청에 S3 서명이 포함되지 않은 경우 와일드카드 문자(*)를 보안 주체에 지정하여 익명 액세스가 허용됩니다. 기본적으로 계정 루트만 해당 계정이 소유한 리소스에 액세스할 수 있습니다. 버킷 정책에서 Principal 요소만 지정하면 됩니다. 그룹 정책의 경우 정책이 연결된 그룹이 암시적 Principal 요소입니다.
리소스/NotResource입니다	Resource 요소는 버킷 및 오브젝트를 식별합니다. ARN(Amazon Resource Name)을 사용하여 리소스를 식별하는 버킷 및 객체에 대한 권한을 허용하거나 거부할 수 있습니다.
작업/NotAction	Action 및 Effect 요소는 권한의 두 구성 요소입니다. 그룹이 리소스를 요청하면 리소스에 대한 액세스가 부여되거나 거부됩니다. 명시적으로 권한을 할당하지 않는 한 액세스가 거부되지만 명시적 DENY를 사용하여 다른 정책이 부여한 권한을 재정의할 수 있습니다.
조건	Condition 요소는 선택 요소입니다. 조건을 사용하면 식을 만들어 정책을 적용해야 하는 시기를 결정할 수 있습니다.

Action 요소에서 와일드카드 문자(*)를 사용하여 모든 작업이나 작업의 하위 집합을 지정할 수 있습니다. 예를 들어 이

작업은 S3:GetObject , S3:PutObject 및 S3:DeleteObject 와 같은 사용 권한을 일치시킵니다.

```
s3:*Object
```

Resource 요소에서 와일드카드 문자(\ **)** 및 **(?)**를 사용할 수 있습니다**.** 별표**()**가 0개 이상의 문자와 일치하면 물음표(?)가 모든 단일 문자와 일치합니다.

Principal 요소에서 모든 사용자에게 권한을 부여하는 익명 액세스를 설정하는 것 외에는 와일드카드 문자는 지원되지 않습니다. 예를 들어 와일드카드(*)를 Principal 값으로 설정합니다.

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

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

다음 예제에서는 Effect , Principal , Action 및 Resource 요소를 사용합니다. 이 예제에서는 "Allow" 효과를 사용하여 Principals, 즉 admin 그룹에 제공하는 전체 버킷 정책 문을 보여 줍니다 `federated-group/admin` 재무그룹을 의미합니다 `federated-group/finance`, 작업 수행 권한 `s3:ListBucket` 을(를) 버킷에 표시합니다 `mybucket` 및 조치 `s3:GetObject` 버킷에 있는 모든 물체

```
{
  "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:iam:s3:::mybucket",
        "arn:aws:iam:s3:::mybucket/*"
      ]
    }
  ]
}
```

버킷 정책은 크기 제한이 20,480바이트이고 그룹 정책은 크기 제한이 5,120바이트입니다.

정책의 일관성

기본적으로 그룹 정책에 대한 모든 업데이트는 최종적으로 일치합니다. 그룹 정책이 일관되면 정책 캐싱으로 인해 변경 내용이 적용되는 데 15분이 더 걸릴 수 있습니다. 기본적으로 버킷 정책에 대한 모든 업데이트는 매우 일관적입니다.

필요에 따라 버킷 정책 업데이트의 일관성 보장을 변경할 수 있습니다. 예를 들어 사이트 중단 중에 버킷 정책의 변경을 사용할 수 있도록 할 수 있습니다.

이 경우 를 설정할 수 있습니다 `Consistency-Control` PutBucketPolicy 요청의 헤더 또는 Put Bucket 정합성 요청을 사용할 수 있습니다. 버킷 정책의 정합성이 보장되면 정책 캐싱으로 인해 변경 내용이 적용되는 데 8초 더 걸릴 수 있습니다.

> 일시적 상황을 해결하기 위해 일관성을 다른 값으로 설정한 경우 작업을 마치면 버킷 수준 설정을 원래 값으로 다시 설정해야 합니다. 그렇지 않으면 이후의 모든 버킷 요청에 수정된 설정이 사용됩니다.

정책 설명에 **ARN**을 사용합니다

정책 문에서 ARN은 Principal 및 Resource 요소에서 사용됩니다.

- 이 구문을 사용하여 S3 리소스 ARN을 지정합니다.

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

- 이 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:

- 별표(*)를 와일드카드로 사용하여 개체 키 안에 0개 이상의 문자를 일치시킬 수 있습니다.
- 개체 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.

정책에서 리소스를 지정합니다

정책 문에서 Resource 요소를 사용하여 사용 권한이 허용되거나 거부되는 버킷 또는 개체를 지정할 수 있습니다.

- 각 정책 문에는 Resource 요소가 필요합니다. 정책에서 리소스는 요소로 표시됩니다 `Resource`또는 `NotResource` 제외.
- S3 리소스 ARN을 사용하여 리소스를 지정합니다. 예를 들면 다음과 같습니다.

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

- 개체 키 내에서 정책 변수를 사용할 수도 있습니다. 예를 들면 다음과 같습니다.

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

- 리소스 값은 그룹 정책이 생성될 때 아직 존재하지 않는 버킷을 지정할 수 있습니다.

정책에 보안 주체를 지정합니다

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

- 버킷 정책의 각 정책 선언에는 Principal 요소가 포함되어야 합니다. 그룹 정책의 정책 설명은 그룹이 보안 주체로 인식되기 때문에 Principal 요소가 필요하지 않습니다.
- 정책에서 주체는 "Principal" 또는 "NotPrincipal" 요소로 표시됩니다.
- 계정 기반 ID는 ID 또는 ARN을 사용하여 지정해야 합니다.

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

- 이 예에서는 계정 루트 및 계정의 모든 사용자를 포함하는 테넌트 계정 ID 27233906934684427525를 사용합니다.

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

- 계정 루트만 지정할 수 있습니다.

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

- 특정 페더레이션 사용자("Alex")를 지정할 수 있습니다.

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

- 특정 통합 그룹("관리자")을 지정할 수 있습니다.

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

- 익명 보안 주체를 지정할 수 있습니다.

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

- 모호함을 방지하려면 사용자 이름 대신 사용자 UUID를 사용할 수 있습니다.

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

  예를 들어 Alex가 조직과 사용자 이름을 그대로 두고 있다고 가정해 보겠습니다 `Alex` 이(가) 삭제됩니다. 새로운 Alex가 조직에 합류하여 동일한 권한이 할당된 경우 `Alex` 사용자 이름, 새 사용자는 의도하지 않게 원래 사용자에게 부여된 권한을 상속할 수 있습니다.

- Principal 값은 버킷 정책이 생성될 때 아직 존재하지 않는 그룹/사용자 이름을 지정할 수 있습니다.

정책에서 사용 권한을 지정합니다

정책에서 Action 요소는 리소스에 대한 권한을 허용/거부하는 데 사용됩니다. 정책에서 지정할 수 있는 사용 권한 집합이 있으며, 이러한 권한은 "작업" 또는 "NotAction" 요소로 표시됩니다. 각 요소는 특정 S3 REST API 작업에 매핑됩니다.

이 표에는 버킷에 적용되는 사용 권한과 객체에 적용되는 사용 권한이 나열되어 있습니다.

> 이제 Amazon S3는 PutBucketReplication 및 DeleteBucketReplication 작업 모두에 대해 S3:PutReplicationConfiguration 권한을 사용합니다. StorageGRID는 원래 Amazon S3 사양과 일치하는 각 작업에 대해 별도의 권한을 사용합니다.

> 기존 값을 덮어쓰는 데 PUT을 사용할 때 삭제가 수행됩니다.

버킷에 적용되는 권한

권한	**S3 REST API** 작업	**StorageGRID** 사용자 지정
S3:생성 버킷	CreateBucket	예. • 참고 *: 그룹 정책에만 사용합니다.
S3:삭제 버킷	삭제 버킷	

권한	**S3 REST API** 작업	**StorageGRID** 사용자 지정
S3:DeleteBucketMetadataNotification	버킷 메타데이터 알림 구성을 삭제합니다	예
S3:삭제 BucketPolicy	DeleteBuckketPolicy를 참조하십시오	
S3:DeleteReplicationConfiguration	DeleteBuckReplication 을 참조하십시오	예, 삽입 및 삭제 권한을 구분합니다
S3:GetBuckketAcl	GetBuckketAcl	
S3:GetBucketCompliance	버킷 규정 준수 가져오기(더 이상 사용되지 않음)	예
S3:GetBuckketConsistency	버킷 일관성 확보	예
S3:GetBuckketCORS	GetBuckketCors 를 참조하십시오	
S3:GetEncryptionConfiguration	GetBuckketEncryption을 참조하십시오	
S3:GetBucketLastAccessTime	버킷 최종 액세스 시간 가져오기	예
S3:GetBucketLocation	GetBucketLocation 을 참조하십시오	
S3:GetBuckMetadataNotification 을 참조하십시오	Bucket 메타데이터 알림 구성 가져오기	예
S3:GetBucketNotification 을 참조하십시오	GetBuckNotificationConfiguration 을 참조하십시오	
S3:GetBuckketObjectLockConfiguration	GetObjectLockConfiguration 을 참조하십시오	
S3:GetBuckketPolicy를 참조하십시오	GetBuckketPolicy를 참조하십시오	
S3:GetBucketTagging	GetBucketTagging	
S3:GetBucketVersioning	GetBuckketVersioning 을 참조하십시오	
S3:GetLifecycleConfiguration	GetBuckLifecycleConfiguration 을 참조하십시오	
S3:GetReplicationConfiguration	GetBucketReplication 을 참조하십시오	

권한	**S3 REST API** 작업	**StorageGRID** 사용자 지정
S3:ListAllMyBucket	• ListBucket • 스토리지 사용량을 가져옵니다	예, 스토리지 사용량 가져오기. • 참고 *: 그룹 정책에만 사용합니다.
S3:목록 버킷	• ListObjects 를 선택합니다 • 머리버킷 • RestoreObject 를 선택합니다	
S3:ListBuckketMultipartUploads	• ListMultipartUploads 를 참조하십시오 • RestoreObject 를 선택합니다	
S3:목록 BuckketVersions	버킷 버전 가져오기	
S3: PutBucketCompliance	버킷 규정 준수(폐기됨)	예
S3: PutBucketConsistency	버킷 일관성을 유지합니다	예
S3: PutBucketCORS	• DeleteBucketCors † • BuckketCors의	
S3:PutEncryptionConfiguration	• DeleteBucketEncryption • PutBucketEncryption을 참조하십시오	
S3:PutBucketLastAccessTime	버킷 최종 접근 시간	예
S3:PutBucketMetadataNotification	Put Bucket 메타데이터 알림 구성	예
S3: PutBucketNotification	PutBucketNotificationConfiguration을 참조하십시오	
S3:PutBucketObjectLockConfiguration	• 를 사용하여 CreateBucket 을 만듭니다 `x-amz-bucket-object-lock-enabled: true` 요청 헤더(S3:CreateBucket 권한도 필요함) • PutObjectLockConfiguration 을 참조하십시오	
S3: PutBucketPolicy	BucketPolicy를 참조하십시오	

권한	**S3 REST API** 작업	**StorageGRID** 사용자 지정
S3: PutBucketTagging	• DeleteBucketTagging † 를 참조하십시오 • BucketTagging	
S3: PutBucketVersioning	PutBucketVersioning을 참조하십시오	
S3: PutLifecycleConfiguration	• DeleteBucketLifecycle † 을 참조하십시오 • PutBucketLifecycleConfiguration을 참조하십시오	
S3:PutReplicationConfiguration	PutBucketReplication을 참조하십시오	예, 삽입 및 삭제 권한을 구분합니다

객체에 적용되는 권한

권한	**S3 REST API** 작업	**StorageGRID** 사용자 지정
S3:중단멀티업로드입니다	• AbortMultipartUpload 를 클릭합니다 • RestoreObject 를 선택합니다	
S3:BypassGovernanceRetention	• DeleteObject 를 클릭합니다 • DeleteObjects 를 클릭합니다 • PutObjectRetention	
S3:DeleteObject 를 선택합니다	• DeleteObject 를 클릭합니다 • DeleteObjects 를 클릭합니다 • RestoreObject 를 선택합니다	
S3:삭제 ObjectTagging	DeleteObjectTagging 을 선택합니다	
S3:DeleteObjectVersionTagging	DeleteObjectTagging(개체의 특정 버전)	
S3:DeleteObjectVersion	DeleteObject(개체의 특정 버전)	
S3:GetObject	• GetObject 를 참조하십시오 • HeadObject 를 선택합니다 • RestoreObject 를 선택합니다 • SelectObjectContent 를 선택합니다	

권한	**S3 REST API** 작업	**StorageGRID** 사용자 지정
S3:GetObjectAcl	GetObjectAcl	
S3:GetObjectLegalHold	GetObjectLegalHold 를 참조하십시오	
S3:GetObjectRetention	GetObjectRetention을 참조하십시오	
S3:GetObjectTagging	GetObjectTagging	
S3:GetObjectVersionTagging	GetObjectTagging(개체의 특정 버전)	
S3:GetObjectVersion	GetObject(개체의 특정 버전)	
S3:ListMultipartUploadParts(S3:ListMultimpartUploadParts) 를	ListParts, RestoreObject 를 참조하십시오	
S3:PutObject	• PutObject 를 선택합니다 • CopyObject 를 선택합니다 • RestoreObject 를 선택합니다 • CreateMultptUpload 를 클릭합니다 • CompleteMultipartUpload를 클릭합니다 • 업로드 파트 • 업로드파트 복사	
S3:PutObjectLegalHold	PutObjectLegalHold를 선택합니다	
S3:PutObjectRetention	PutObjectRetention	
S3:PutObjectTagging	PutObjectTagging	
S3:PutObjectVersionTagging	PutObjectTagging(개체의 특정 버전)	
S3:PutOverwriteObject	• PutObject 를 선택합니다 • CopyObject 를 선택합니다 • PutObjectTagging • DeleteObjectTagging 을 선택합니다 • CompleteMultipartUpload를 클릭합니다	예
S3:RestoreObject	RestoreObject 를 선택합니다	

PutOverwriteObject 권한을 사용합니다

S3:PutOverwriteObject 권한은 개체를 만들거나 업데이트하는 작업에 적용되는 사용자 지정 StorageGRID 권한입니다. 이 사용 권한의 설정에 따라 클라이언트가 개체의 데이터, 사용자 정의 메타데이터 또는 S3 오브젝트 태그 지정을 덮어쓸 수 있는지 여부가 결정됩니다.

이 권한에 사용할 수 있는 설정은 다음과 같습니다.

- * 허용 *: 클라이언트가 개체를 덮어쓸 수 있습니다. 기본 설정입니다.
- * 거부 *: 클라이언트가 개체를 덮어쓸 수 없습니다. Deny 로 설정된 경우 PutOverwriteObject 권한은 다음과 같이 작동합니다.
  - 기존 객체가 같은 경로에 있는 경우:
    - 오브젝트의 데이터, 사용자 정의 메타데이터 또는 S3 오브젝트 태깅을 덮어쓸 수 없습니다.
    - 진행 중인 모든 수집 작업이 취소되고 오류가 반환됩니다.
    - S3 버전 관리를 사용하는 경우 거부 설정을 사용하면 PutObjectTagging 또는 DeleteObjectTagging 작업에서 개체 및 해당 비최신 버전에 대한 TagSet을 수정할 수 없습니다.
  - 기존 개체를 찾을 수 없으면 이 권한은 적용되지 않습니다.
- 이 권한이 없으면 Allow가 설정된 것과 효과가 같습니다.

> 현재 S3 정책이 덮어쓰기를 허용하고 PutOverwriteObject 권한이 Deny 로 설정된 경우 클라이언트는 개체의 데이터, 사용자 정의 메타데이터 또는 개체 태그를 덮어쓸 수 없습니다. 또한 * 클라이언트 수정 방지 * 확인란이 선택된 경우(* 구성 * > * 보안 설정 * > * 네트워크 및 개체 *) 해당 설정은 PutOverwriteObject 권한 설정을 재정의합니다.

정책에서 조건을 지정합니다

조건은 정책이 적용되는 시점을 정의합니다. 조건은 연산자 및 키 값 쌍으로 구성됩니다.

조건은 평가에 키 값 쌍을 사용합니다. 조건 요소에는 여러 조건이 포함될 수 있으며 각 조건에는 여러 키 값 쌍이 포함될 수 있습니다. 조건 블럭은 다음 형식을 사용합니다:

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

다음 예제에서 IPAddress 조건은 SOURCEIP 조건 키를 사용합니다.

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

조건 연산자는 다음과 같이 분류됩니다.

- 문자열
- 숫자
- 부울
- IP 주소입니다
- Null 확인

조건 연산자	설명
StringEquals	정확한 일치(대/소문자 구분)를 기반으로 문자열 값과 키를 비교합니다.
StringNotEquals	키를 부정 일치(대/소문자 구분)를 기반으로 문자열 값과 비교합니다.
StringEqualsIgnoreCase 를 참조하십시오	정확한 일치를 기준으로 문자열 값과 키를 비교합니다(대/소문자 무시).
StringNotEqualsIgnoreCase 를 참조하십시오	Negated matching (대소문자 무시)을 기준으로 문자열 값과 키를 비교합니다.
StringLike 를 선택합니다	정확한 일치(대/소문자 구분)를 기반으로 문자열 값과 키를 비교합니다. 및 * 를 포함할 수 있습니까? 와일드카드 문자.
StringNotLike 를 참조하십시오	키를 부정 일치(대/소문자 구분)를 기반으로 문자열 값과 비교합니다. 및 * 를 포함할 수 있습니까? 와일드카드 문자.
NumericEquals	정확한 일치를 기준으로 키를 숫자 값과 비교합니다.
NumericNotEquals	키를 부정 일치를 기준으로 숫자 값과 비교합니다.
NumericGreaterThan	"보다 큼" 일치를 기준으로 키를 숫자 값과 비교합니다.
NumericGreaterThanEquals	"보다 크거나 같음" 일치를 기준으로 키를 숫자 값과 비교합니다.
NumericLessThan	"보다 작음" 일치를 기준으로 키를 숫자 값과 비교합니다.
NumericLessThanEquals	"보다 작거나 같음" 일치를 기준으로 키를 숫자 값과 비교합니다.
불입니다	"true 또는 false" 일치를 기준으로 키를 부울 값과 비교합니다.
IP 주소	키를 IP 주소 또는 IP 주소 범위와 비교합니다.

조건 연산자	설명
NotIpAddress 를 참조하십시오	부정 일치를 기준으로 IP 주소 또는 IP 주소 범위와 키를 비교합니다.
null입니다	현재 요청 컨텍스트에 조건 키가 있는지 확인합니다.

지원되는 조건 키

상태 키	작업	설명
AWS: SOURCEIP	IP 연산자	요청이 전송된 IP 주소와 비교합니다. 버킷 또는 오브젝트 작업에 사용할 수 있습니다. • 참고: * S3 요청이 관리 노드 및 게이트웨이 노드의 로드 밸런서 서비스를 통해 전송된 경우 로드 밸런서 서비스의 IP 주소 업스트림과 비교됩니다. • 참고 *: 타사, 비투명 로드 밸런서가 사용되는 경우 이 로드 밸런서의 IP 주소와 비교합니다. 모두 `X-Forwarded-For` 헤더의 유효성을 확인할 수 없기 때문에 헤더가 무시됩니다.
AWS: 사용자 이름	리소스/ID입니다	요청이 전송된 보낸 사람의 사용자 이름과 비교합니다. 버킷 또는 오브젝트 작업에 사용할 수 있습니다.
S3: 구분 기호	S3:ListBucket 및 S3: ListBuckketVersions 권한	는 ListObjects 또는 ListObjectVersions 요청에 지정된 구분 기호 매개 변수와 비교됩니다.

상태 키	작업	설명
S3: ExistingObjectTag/<tag-key>	S3:삭제 ObjectTagging S3:DeleteObjectVersionTagging S3:GetObject S3:GetObjectAcl 3: GetObjectTagging S3:GetObjectVersion S3:GetObjectVersionAcl S3:GetObjectVersionTagging S3: PutObjectAcl S3:PutObjectTagging S3: PutObjectVersionAcl S3:PutObjectVersionTagging	기존 개체에 특정 태그 키와 값이 있어야 합니다.
S3: 최대 키	S3:ListBucket 및 S3: ListBuckketVersions 권한	는 ListObjects 또는 ListObjectVersions 요청에 지정된 max-keys 매개 변수와 비교됩니다.
S3: 오브젝트 잠금 장치 - 남은 보존 기간(일)	S3:PutObject	에 지정된 보존 종료 날짜와 비교합니다 `x-amz-object-lock-retain-until-date` 다음 요청에 대해 허용 범위 내에 있는지 확인하기 위해 버킷 기본 보존 기간에서 헤더를 요청하거나 계산합니다. • PutObject 를 선택합니다 • CopyObject 를 선택합니다 • CreateMultptUpload 를 클릭합니다
S3: 오브젝트 잠금 장치 - 남은 보존 기간(일)	S3:PutObjectRetention	PutObjectRetention 요청에 지정된 유지 종료 날짜와 비교하여 허용 범위 내에 있는지 확인합니다.
S3: 접두어	S3:ListBucket 및 S3: ListBuckketVersions 권한	는 ListObjects 또는 ListObjectVersions 요청에 지정된 접두사 매개 변수와 비교됩니다.

상태 키	작업	설명
S3: RequestObjectTag/<tag-key>	S3:PutObject S3:PutObjectTagging S3:PutObjectVersionTagging	개체 요청에 태그가 포함된 경우 특정 태그 키와 값이 필요합니다.

정책에 변수를 지정합니다

정책의 변수를 사용하여 사용 가능한 정책 정보를 채울 수 있습니다. 에서 정책 변수를 사용할 수 있습니다 `Resource` 의 요소 및 문자열 비교 `Condition` 요소.

이 예제에서 변수는 입니다 `${aws:username}` 은(는) Resource 요소의 일부입니다.

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

이 예제에서 변수는 입니다 `${aws:username}` 조건 블럭의 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.
`${s3:prefix}`	서비스별 prefix key를 제공된 variable 로 사용한다.
`${s3:max-keys}`	서비스별 최대 키 키를 제공된 변수로 사용합니다.
`${*}`	특수 문자. 문자를 리터럴 * 문자로 사용합니다.
`${?}`	특수 문자. 문자를 리터럴로 사용합니까? 문자.
`${$}`	특수 문자. 문자를 리터럴 $ 문자로 사용합니다.

특별한 처리가 필요한 정책을 생성합니다

때로는 정책에 따라 보안이 위험하거나 계정 루트 사용자를 잠그는 등 지속적인 작업에 위험한 사용 권한을 부여할 수 있습니다. StorageGRID S3 REST API 구현은 Amazon보다 정책 검증 중에 덜 제한적이지만 정책 평가 중에도 동일하게 엄격합니다.

정책 설명입니다	정책 유형입니다	아마존 행동	**StorageGRID** 동작
루트 계정에 대한 모든 권한을 스스로 거부합니다	버킷	유효하고 적용되지만 루트 사용자 계정은 모든 S3 버킷 정책 작업에 대한 권한을 보유합니다	동일합니다
사용자/그룹에 대한 모든 권한을 스스로 거부합니다	그룹	유효하고 시행되었습니다	동일합니다
외부 계정 그룹에 모든 권한을 허용합니다	버킷	주체가 잘못되었습니다	유효하지만 모든 S3 버킷 정책 작업에 대한 권한은 정책에서 허용하는 경우 405 메서드 허용되지 않음 오류를 반환합니다
외부 계정 루트 또는 사용자에게 모든 권한을 허용합니다	버킷	유효하지만 모든 S3 버킷 정책 작업에 대한 권한은 정책에서 허용하는 경우 405 메서드 허용되지 않음 오류를 반환합니다	동일합니다
모든 사용자에게 모든 작업에 대한 사용 권한을 허용합니다	버킷	유효하지만 모든 S3 버킷 정책 작업에 대한 사용 권한이 외국 계정 루트 및 사용자에 대해 405 메서드 허용 안 됨 오류를 반환합니다	동일합니다
모든 작업에 대한 모든 사용자의 권한을 거부합니다	버킷	유효하고 적용되지만 루트 사용자 계정은 모든 S3 버킷 정책 작업에 대한 권한을 보유합니다	동일합니다
보안 주체는 존재하지 않는 사용자 또는 그룹입니다	버킷	주체가 잘못되었습니다	유효합니다
리소스가 존재하지 않는 S3 버킷입니다	그룹	유효합니다	동일합니다
보안 주체는 로컬 그룹입니다	버킷	주체가 잘못되었습니다	유효합니다

정책 설명입니다	정책 유형입니다	아마존 행동	**StorageGRID** 동작
정책은 개체를 넣을 수 있는 비소유자 계정(익명 계정 포함) 권한을 부여합니다.	버킷	유효합니다. 객체는 생성자 계정이 소유하며 버킷 정책은 적용되지 않습니다. 생성자 계정은 개체 ACL을 사용하여 개체에 대한 액세스 권한을 부여해야 합니다.	유효합니다. 오브젝트는 버킷 소유자 계정이 소유합니다. 버킷 정책이 적용됩니다.

WORM(Write-Once-Read-Many) 보호

WORM(Write-Once-Read-Many) 버킷을 생성하여 데이터, 사용자 정의 오브젝트 메타데이터 및 S3 오브젝트 태깅을 보호할 수 있습니다. 새 객체를 생성하고 기존 컨텐츠를 덮어쓰거나 삭제하지 못하도록 WORM 버킷을 구성합니다. 여기에 설명된 방법 중 하나를 사용합니다.

덮어쓰기가 항상 거부되도록 하려면 다음을 수행할 수 있습니다.

- Grid Manager에서 * 구성 * > * 보안 * > * 보안 설정 * > * 네트워크 및 개체 * 로 이동하여 * 클라이언트 수정 방지 * 확인란을 선택합니다.
- 다음 규칙 및 S3 정책을 적용합니다.
  - S3 정책에 PutOverwriteObject 거부 작업을 추가합니다.
  - DeleteObject 거부 작업을 S3 정책에 추가합니다.
  - S3 정책에 PutObject 허용 작업을 추가합니다.

> S3 정책에서 DeleteObject를 DENY로 설정해도 "30일 후 복사본 제로" 같은 규칙이 있을 때 ILM이 개체를 삭제할 수 없습니다.

> 이러한 규칙과 정책이 모두 적용되더라도 동시 쓰기를 방지하지 않습니다(상황 A 참조). 순차적 완료된 덮어쓰기를 방지합니다(상황 B 참조).

- 상황 A *: 동시 쓰기(보호 안 됨)

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

- 상황 B *: 순차적 완료된 덮어쓰기(방지됨)

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

관련 정보

- "StorageGRID ILM 규칙이 개체를 관리하는 방법"

- "버킷 정책의 예"
- "그룹 정책의 예"
- "ILM을 사용하여 개체를 관리합니다"
- "테넌트 계정을 사용합니다"

버킷 정책의 예

이 섹션의 예를 사용하여 버킷에 대한 StorageGRID 액세스 정책을 구축합니다.

버킷 정책은 정책이 연결된 버킷에 대한 액세스 권한을 지정합니다. 버킷 정책은 S3 PutBucketPolicy API를 사용하여 구성됩니다. 을 참조하십시오 "버킷 작업".

다음 명령에 따라 AWS CLI를 사용하여 버킷 정책을 구성할 수 있습니다.

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

예: 모든 사용자가 버킷에 읽기 전용 액세스를 허용합니다

이 예제에서는 anonymous를 비롯한 모든 사용자가 버킷의 오브젝트를 나열하고 버킷의 모든 오브젝트에 대해 GetObject 작업을 수행할 수 있습니다. 다른 모든 작업은 거부됩니다. 이 정책은 계정 루트 외에는 버킷에 쓸 수 있는 권한이 없으므로 특히 유용하지 않을 수 있습니다.

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

예: 한 계정의 모든 사용자가 완전히 액세스할 수 있도록 허용하고 다른 계정의 모든 사용자는 버킷에 읽기 전용으로 액세스할 수 있습니다

이 예제에서는 지정된 계정의 모든 사용자가 버킷에 완전히 액세스할 수 있지만, 지정된 다른 계정의 모든 사용자는 버킷을 나열하고 으로 시작하는 버킷의 개체에 대해 GetObject 작업만 수행할 수 있습니다 `shared/` 개체 키 접두사입니다.

> StorageGRID에서 비소유자 계정(익명 계정 포함)으로 생성된 객체는 버킷 소유자 계정이 소유합니다. 버킷 정책은 이러한 오브젝트에 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/*"
        }
      }
    }
  ]
}
```

예: 모든 사용자가 버킷에 대한 읽기 전용 액세스 및 지정된 그룹에 의한 전체 액세스 허용

이 예제에서는 anonymous를 포함하는 모든 사용자가 버킷에 있는 모든 오브젝트에 대해 버킷을 나열하고 GetObject 작업을 수행할 수 있으며, 이 작업은 그룹에 속한 사용자만 수행할 수 있습니다 `Marketing` 지정된 계정에서 전체 액세스가 허용됩니다.

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

예: 클라이언트가 **IP** 범위에 있는 경우 모든 사용자가 버킷에 대한 읽기 및 쓰기 액세스를 허용합니다

이 예제에서는 요청이 지정된 IP 범위(54.240.143.0 ~ 54.240.143.255, 54.240.143.188 제외)에서 발생한 경우 anonymous를 포함한 모든 사람이 버킷을 나열하고 버킷의 모든 오브젝트에 대해 오브젝트 작업을 수행할 수 있습니다. 다른 모든 작업이 거부되고 IP 범위를 벗어난 모든 요청이 거부됩니다.

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

예: 지정된 통합 사용자가 단독으로 버킷을 완전히 액세스할 수 있도록 허용합니다

이 예에서는 페더레이션 사용자 Alex가 에 대한 전체 액세스를 허용합니다 `examplebucket` 버킷과 그 물체. "root"를 포함한 다른 모든 사용자는 모든 작업을 명시적으로 거부합니다. 그러나 "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/*"
      ]
    }
  ]
}
```

예**: PutOverwriteObject** 권한

이 예에서 는 입니다 `Deny` PutOverwriteObject 및 DeleteObject 에 대한 효과 개체의 데이터, 사용자 정의 메타데이터 및 S3 개체 태그 지정을 덮어쓰거나 삭제할 수 없습니다.

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

그룹 정책의 예

이 섹션의 예제를 사용하여 그룹에 대한 StorageGRID 액세스 정책을 작성합니다.

그룹 정책은 정책이 연결된 그룹에 대한 액세스 권한을 지정합니다. 아니요 `Principal` 암시적 정책이므로 정책의 요소입니다. 그룹 정책은 테넌트 관리자 또는 API를 사용하여 구성됩니다.

예: 테넌트 관리자를 사용하여 그룹 정책을 설정합니다

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

- * S3 액세스 없음 *: 기본 옵션. 버킷 정책을 통해 액세스 권한이 부여되지 않은 한 이 그룹의 사용자는 S3 리소스에 액세스할 수 없습니다. 이 옵션을 선택하면 루트 사용자만 기본적으로 S3 리소스에 액세스할 수 있습니다.
- * 읽기 전용 액세스 *: 이 그룹의 사용자는 S3 리소스에 대한 읽기 전용 액세스 권한을 가집니다. 예를 들어 이 그룹의 사용자는 개체를 나열하고 개체 데이터, 메타데이터 및 태그를 읽을 수 있습니다. 이 옵션을 선택하면 읽기 전용 그룹 정책의 JSON 문자열이 텍스트 상자에 나타납니다. 이 문자열을 편집할 수 없습니다.
- * 전체 액세스 *: 이 그룹의 사용자는 버킷을 포함하여 S3 리소스에 대한 모든 액세스 권한을 가집니다. 이 옵션을 선택하면 전체 액세스 그룹 정책의 JSON 문자열이 텍스트 상자에 나타납니다. 이 문자열을 편집할 수 없습니다.
- * 랜섬웨어 완화 *: 이 샘플 정책은 이 테넌트의 모든 버킷에 적용됩니다. 이 그룹의 사용자는 일반적인 작업을 수행할 수 있지만 개체 버전 관리가 활성화된 버킷에서 개체를 영구적으로 삭제할 수는 없습니다.

  모든 버킷 관리 권한이 있는 테넌트 관리자 사용자는 이 그룹 정책을 재정의할 수 있습니다. 모든 버킷 관리 권한을 신뢰할 수 있는 사용자로 제한하고 가능한 경우 MFA(Multi-Factor Authentication)를 사용합니다.

- * 사용자 정의 *: 그룹의 사용자에게는 텍스트 상자에 지정한 사용 권한이 부여됩니다.

예: 모든 버킷에 대한 그룹 전체 액세스 허용

이 예에서 그룹의 모든 구성원은 버킷 정책에 의해 명시적으로 거부되지 않는 한 테넌트 계정이 소유한 모든 버킷에 대해 전체 액세스가 허용됩니다.

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

예: 모든 버킷에 대한 그룹 읽기 전용 액세스를 허용합니다

이 예제에서 그룹의 모든 구성원은 버킷 정책에 의해 명시적으로 거부되지 않는 한 S3 리소스에 대해 읽기 전용 액세스 권한을 갖습니다. 예를 들어 이 그룹의 사용자는 개체를 나열하고 개체 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:::*"
    }
  ]
}
```

예: 그룹 구성원이 버킷의 "폴더"에만 모든 권한을 부여할 수 있습니다

이 예제에서 그룹의 구성원은 지정된 버킷의 특정 폴더(키 접두사)를 나열하고 액세스할 수만 있습니다. 이러한 폴더의 개인 정보를 확인할 때는 다른 그룹 정책 및 버킷 정책의 액세스 권한을 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}/*"
    }
  ]
}
```

감사 로그에서 **S3** 작업을 추적했습니다

감사 메시지는 StorageGRID 서비스에서 생성되고 텍스트 로그 파일에 저장됩니다. 감사 로그에서 S3별 감사 메시지를 검토하여 버킷 및 오브젝트 작업에 대한 세부 정보를 확인할 수 있습니다.

감사 로그에서 버킷 작업을 추적했습니다

- CreateBucket
- 삭제 버킷
- 삭제 BucketTagging
- DeleteObjects 를 클릭합니다
- GetBucketTagging
- 머리버킷
- ListObjects 를 선택합니다
- ListObjectVersions 를 선택합니다
- 버킷 규정 준수
- BucketTagging
- PutBucketVersioning을 참조하십시오

감사 로그에서 추적된 객체 작업입니다

- CompleteMultipartUpload를 클릭합니다
- CopyObject 를 선택합니다
- DeleteObject 를 클릭합니다
- GetObject 를 참조하십시오
- HeadObject 를 선택합니다
- PutObject 를 선택합니다
- RestoreObject 를 선택합니다
- 개체 를 선택합니다
- UploadPart(ILM 규칙이 Balanced 또는 Strict 수집을 사용하는 경우)
- UploadPartCopy(ILM 규칙이 Balanced 또는 Strict 수집을 사용하는 경우)

관련 정보

- "감사 로그 파일에 액세스합니다"
- "클라이언트가 감사 메시지를 기록합니다"
- "클라이언트가 감사 메시지를 읽습니다"