Amazon S3 REST API 지원

StorageGRID software

NetApp
July 23, 2026

목차

Amazon S3 REST API 지원

StorageGRID에서 지원되는 S3 REST API 작업 및 제한 사항

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

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

날짜 처리

StorageGRID에서 구현한 S3 REST API는 유효한 HTTP 날짜 형식만 지원합니다.

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

일반적인 요청 헤더

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

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

일반적인 응답 헤더

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

응답 헤더	구현
x-amz-id-2	사용하지 않음

StorageGRID S3 REST API가 요청을 인증하는 방법

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

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

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

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

HTTP Authorization 헤더 사용

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

쿼리 매개변수 사용

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

StorageGRID에서 지원되는 서비스 작업

StorageGRID 시스템은 서비스에 대해 다음과 같은 작업을 지원합니다.

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

StorageGRID의 S3 버킷 작업 및 구현 세부 정보

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

각 그리드는 최대 100,000개의 버킷을 가질 수 있습니다.

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

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

자세한 내용은 다음을 참조하십시오.

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

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

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

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

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

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

작업	구현
GetBucketVersioning	이 구현에서는 `versioning` 하위 리소스를 사용하여 버킷의 버전 관리 상태를 반환합니다. • *blank*: 버전 관리가 활성화된 적이 없습니다(버킷이 "Unversioned"입니다). • 활성화됨: 버전 관리가 활성화되었습니다. • 일시 중단됨: 이전에 버전 관리가 활성화되었으나 현재 일시 중단되었습니다.
GetObjectLockConfiguration	구성된 경우 버킷의 기본 보존 모드와 기본 보존 기간을 반환합니다. "S3 REST API를 사용하여 S3 Object Lock 구성"을 참조하십시오.
HeadBucket	버킷이 존재하는지, 그리고 해당 버킷에 액세스할 권한이 있는지 확인합니다. 이 연산의 결과는 다음과 같습니다. • `x-ntap-sg-bucket-id`: 버킷의 UUID(UUID 형식)입니다. • `x-ntap-sg-trace-id`: 해당 요청의 고유 추적 ID입니다.
ListObjects 및 ListObjectsV2 (이전 명칭: GET Bucket)	버킷에 있는 객체 중 일부 또는 전부(최대 1,000개)를 반환합니다. 객체의 스토리지 클래스는 해당 객체가 `REDUCED_REDUNDANCY` 스토리지 클래스 옵션을 사용하여 수집된 경우에도 두 가지 값 중 하나를 가질 수 있습니다. • `STANDARD`, 이는 해당 객체가 스토리지 노드로 구성된 스토리지 풀에 저장되어 있음을 나타냅니다. • `GLACIER`이는 해당 객체가 클라우드 스토리지 풀에서 지정한 외부 버킷으로 이동되었음을 나타냅니다. 버킷에 동일한 접두사를 가진 삭제된 키가 대량으로 포함되어 있는 경우, 응답에 키가 포함되지 않은 `CommonPrefixes`이 일부 포함될 수 있습니다. HeadObject 및 ListObject 요청의 경우 StorageGRID는 LastModified 타임스탬프를 서로 다른 정밀도로 반환하는 반면 AWS는 다음 예에서와 같이 동일한 정밀도로 타임스탬프를 반환합니다. • StorageGRID HeadObject: "LastModified": "2024-09-26T16:43:24+00:00" • StorageGRID ListObject: "LastModified": "2024-09-26T16:43:24.931000+00:00" • AWS HeadObject: "LastModified": "2023-10-17T00:19:54+00:00" • AWS ListObject: "LastModified": "2023-10-17T00:19:54+00:00"
ListObjectVersions (이전 명칭: GET Bucket Object versions)	버킷에 대한 읽기 권한이 있는 경우, `versions` 하위 리소스와 함께 이 작업을 사용하면 버킷에 있는 모든 객체 버전의 메타데이터가 나열됩니다.

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

작업	구현

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

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

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

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

객체에 대한 연산

StorageGRID에서 객체에 대한 S3 REST API 작업을 구현하는 방법

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

다음 조건은 모든 객체 작업에 적용됩니다.

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

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

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

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

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

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

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

StorageGRID에서 지원되는 Amazon S3 Select 작업

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

> 목록에 없는 항목은 지원되지 않습니다.

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

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

조항

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

데이터 유형

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

운영자

논리 연산자

- AND
- NOT
- 또는

비교 연산자

- <
- >
- ⇐
- >=
- =

- =
- <>
- !=
- 사이
- IN

패턴 매칭 연산자

- LIKE
- _
- %

단일 연산자

- IS NULL
- NULL이 아닙니다

수학 연산자

- +
- -
- *
- /
- %

StorageGRID는 Amazon S3 Select 연산자 우선순위를 따릅니다.

집계 함수

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

조건 함수

- 사례
- COALESCE
- NULLIF

변환 함수

- CAST(지원되는 데이터 형식의 경우)

날짜 함수

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

문자열 함수

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

StorageGRID에서 서버 측 암호화를 사용하는 방법

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

${post_edited_translations.segment}

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

  ${post_edited_translations.segment}

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

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

${post_edited_translations.segment}

${post_edited_translations.segment}

`x-amz-server-side-encryption`

SSE 요청 헤더는 다음 객체 작업에서 지원됩니다.

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

SSE-C 사용

${post_edited_translations.segment}

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

${post_edited_translations.segment}

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

${post_edited_translations.segment}

SSE-C를 사용하기 전에 다음 사항을 고려하십시오.

- https를 사용해야 합니다.

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

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

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

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

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

관련 정보

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

S3 CopyObject 요청을 사용하여 StorageGRID에 객체의 복사본을 생성합니다

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

충돌 해결

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

객체 크기

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

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

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

사용자 메타데이터의 **UTF-8** 문자

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

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

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

지원되는 요청 헤더

다음과 같은 요청 헤더가 지원됩니다.

- `Content-Type`
- `x-amz-copy-source`
- `x-amz-copy-source-if-match`
- `x-amz-copy-source-if-none-match`
- `x-amz-copy-source-if-unmodified-since`
- `x-amz-copy-source-if-modified-since`
- `x-amz-meta-`, 그 뒤에 사용자 정의 메타데이터를 포함하는 이름-값 쌍이 옵니다
- `x-amz-metadata-directive`: 기본값은 `COPY`이며, 이를 사용하면 객체와 관련 메타데이터를 복사할 수 있습니다.

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

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

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

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

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

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

    서버 측 암호화에 대한 요청 헤더을 참조하십시오

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않습니다.

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

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

- `If-None-Match`

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

- `x-amz-checksum-algorithm`

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

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

스토리지 클래스 옵션

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

- `STANDARD`

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

- `REDUCED_REDUNDANCY`

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

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

CopyObject에서 x-amz-copy-source 사용

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

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

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

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

서버 측 암호화에 대한 요청 헤더

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

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

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

- StorageGRID(SSE)에서 관리하는 고유 키로 대상 객체(복사본)를 암호화하려면 CopyObject 요청에 다음 헤더를 포함하십시오.
  - `x-amz-server-side-encryption`

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

버전 관리

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

GetObject 요청을 사용하여 StorageGRID의 버킷에서 객체 검색

S3 GetObject 요청을 사용하여 S3 버킷에서 객체를 검색할 수 있습니다.

GetObject 및 멀티파트 객체

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

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

사용자 메타데이터의 **UTF-8** 문자

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

지원되는 요청 헤더

다음 요청 헤더가 지원됩니다.

- `x-amz-checksum-mode`: 지정 `ENABLED`

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

지원되지 않는 요청 헤더

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

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

버전 관리

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

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

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

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

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

클라우드 스토리지 풀 객체에 대한 **GetObject**의 동작

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

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

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

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

클라우드 스토리지 풀의 멀티파트 또는 분할된 객체

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

이러한 경우:

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

GetObject 및 그리드 간 복제

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

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

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

S3 HeadObject 요청을 사용하여 StorageGRID의 객체에서 메타데이터를 검색합니다

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

HeadObject 및 멀티파트 객체

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

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

사용자 메타데이터의 **UTF-8** 문자

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

지원되는 요청 헤더

다음 요청 헤더가 지원됩니다.

- `x-amz-checksum-mode`

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

지원되지 않는 요청 헤더

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

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

버전 관리

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

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

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

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

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

HeadObject 클라우드 스토리지 풀 객체에 대한 응답

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

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

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

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

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

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

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

객체의 상태	**HeadObject**에 대한 응답

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

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

클라우드 스토리지 풀의 멀티파트 또는 분할된 객체

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

HeadObject 및 그리드 간 복제

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

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

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

S3 PutObject 요청을 사용하여 StorageGRID의 버킷에 객체 추가

S3 PutObject 요청을 사용하여 버킷에 객체를 추가할 수 있습니다.

충돌 해결

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

객체 크기

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

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

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

사용자 메타데이터 크기

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

사용자 메타데이터의 **UTF-8** 문자

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

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

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

객체 태그 제한

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

객체 소유권

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

생성한 객체도 포함됩니다.

지원되는 요청 헤더

다음과 같은 요청 헤더가 지원됩니다.

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

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

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

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

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

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

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

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

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

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

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

    이러한 헤더 없이 요청이 이루어지면 버킷의 기본 보존 설정이 객체 버전 모드와 보존 만료일을 계산하는 데 사용됩니다. "S3 REST API를 사용하여 S3 Object Lock 구성"을 참조하십시오.
- SSE 요청 헤더:
  - `x-amz-server-side-encryption`
  - `x-amz-server-side-encryption-customer-key-MD5`
  - `x-amz-server-side-encryption-customer-key`
  - `x-amz-server-side-encryption-customer-algorithm`

    서버 측 암호화에 대한 요청 헤더을 참조하십시오

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않습니다.

- `If-Match`

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

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

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

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

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

다음 값을 사용할 수 있습니다 x-amz-storage-class:

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

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

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

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

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

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

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

서버 측 암호화에 대한 요청 헤더

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

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

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

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

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

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

버전 관리

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

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

Authorization 헤더에 대한 서명 계산

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

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

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

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

관련 정보

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

S3 RestoreObject 요청을 사용하여 StorageGRID에서 객체 복원

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

지원되는 요청 유형

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

버전 관리

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

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

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

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

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

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

S3 SelectObjectContent 요청을 사용하여 StorageGRID에서 객체 데이터 필터링

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

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

시작하기 전에

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

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

CSV 요청 구문 예시

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

Parquet 요청 구문 예시

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

SQL 쿼리 예시

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

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

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

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

AWS-CLI 사용 예시 **(CSV)**

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

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

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

AWS-CLI 사용 예시 (Parquet)

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

출력 파일인 changes.csv의 처음 몇 줄은 다음과 같습니다.

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

멀티파트 업로드 작업

StorageGRID에서 지원되는 멀티파트 업로드 작업

이 섹션에서는 StorageGRID가 멀티파트 업로드 작업을 지원하는 방법을 설명합니다.

모든 멀티파트 업로드 작업에는 다음과 같은 조건과 유의사항이 적용됩니다.

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

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

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

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

StorageGRID S3 REST API가 멀티파트 업로드를 완료하는 방법

CompleteMultipartUpload 작업은 이전에 업로드된 부분들을 조합하여 객체의 멀티파트 업로드를 완료합니다.

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

충돌 해결

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

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

지원되는 요청 헤더

다음과 같은 요청 헤더가 지원됩니다.

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

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

- `STANDARD`

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

- `REDUCED_REDUNDANCY`

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

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

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

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

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않습니다.

- `If-Match`

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

- `If-None-Match`

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

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

버전 관리

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

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

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

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

복제, 알림 또는 메타데이터 알림 실패

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

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

"플랫폼 서비스 문제 해결"을 참조하십시오.

StorageGRID의 S3 CreateMultipartUpload 요청 헤더 및 옵션

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

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

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

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

- `STANDARD` (기본값)

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

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

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

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

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

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

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

지원되는 요청 헤더

다음과 같은 요청 헤더가 지원됩니다.

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

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

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

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

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

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

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

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

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

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

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

  "S3 REST API를 사용하여 S3 Object Lock 구성"

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

    서버 측 암호화에 대한 요청 헤더

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

서버 측 암호화에 대한 요청 헤더

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

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

UploadPart 요청)에 이 세 가지 헤더를 모두 사용하십시오.

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

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

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않습니다.

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

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

버전 관리

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

S3 ListMultipartUploads 작업 및 StorageGRID에서 지원되는 매개 변수

ListMultipartUploads 작업은 버킷에 대해 진행 중인 멀티파트 업로드 목록을 표시합니다.

다음 요청 매개변수가 지원됩니다.

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

버전 관리

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

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

UploadPart 작업은 객체의 멀티파트 업로드에서 파트를 업로드합니다.

지원되는 요청 헤더

다음과 같은 요청 헤더가 지원됩니다.

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

서버 측 암호화에 대한 요청 헤더

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

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

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

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

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

지원되지 않는 요청 헤더

다음 요청 헤더는 지원되지 않습니다.

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

버전 관리

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

S3 UploadPartCopy 작업을 사용하여 StorageGRID에 객체의 일부를 업로드합니다

UploadPartCopy 작업은 기존 객체의 데이터를 데이터 소스로 복사하여 객체의 일부를

업로드합니다.

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

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

다음과 같은 요청 헤더가 지원됩니다.

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

서버 측 암호화에 대한 요청 헤더

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

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

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

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

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

버전 관리

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

StorageGRID S3 REST API 오류 응답

StorageGRID 시스템은 적용 가능한 모든 표준 S3 REST API 오류 응답을 지원합니다. 또한

StorageGRID 구현은 몇 가지 사용자 지정 응답을 추가로 제공합니다.

지원되는 **S3 API** 오류 코드

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

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

StorageGRID 사용자 지정 오류 코드

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

저작권 정보

상표 정보